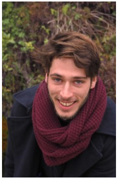
Maximizing Covered Area in the Euclidean Plane with Connectivity Constraint


Chien-Chung Huang **Mathieu Mari**[1] Claire Mathieu Joseph S. B. Mitchell Nabil H. Mustafa

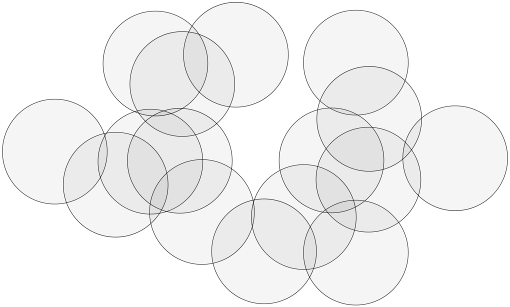
[1]École Normale Supérieure, Université PSL, Paris

Definition: Connected Unit-disk *k*-coverage

In: A (connected) set of unit-area-disks in the Euclidean plane and an integer ***k***

Out: A connected subset *S* of size ***k***

Goal: Maximize the area covered by *S*

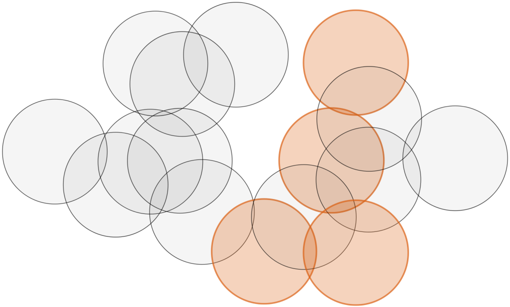


Definition: Connected Unit-disk k-coverage

In: A (connected) set of unit-area-disks in the Euclidean plane and an integer k

Out: A connected subset S of size k

Goal: Maximize the area covered by S



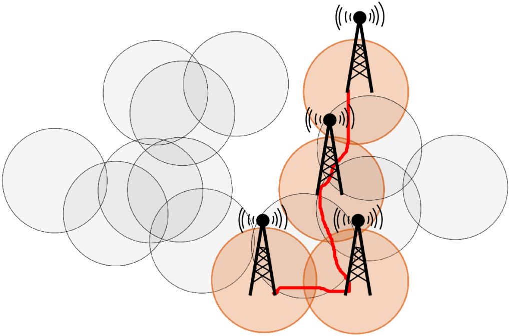
$k = 4$

Definition: Connected Unit-disk *k*-coverage

In: A (connected) set of unit-area-disks in the Euclidean plane and an integer ***k***

Out: A connected subset *S* of size ***k***

Goal: Maximize the area covered by *S*



generalisations

budgeted connected dominating set: $\frac{1}{13}(1-1/e)$-approximation [Khuller, Purohit, Sarpatwar, 2014], very recently improved to $\frac{1}{7}(1-1/e)$? [Lamprou, Sigalas, Zissimopoulos, 2019]

connected *k*-coverage: $\Omega(1/\sqrt{k})$-approximation when objective function is monotone submodular. [Kuo, Lin, Tsai, 2015]

Related results

***k*-coverage:** optimal greedy $1-1/e$ approximation for monotone submodular function.
(f submodular: $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B), \forall A \subseteq B \subseteq X, \forall x \in X$)

unit-disk *k*-coverage: PTAS. [Chaplik, De, Ravsky, Spoerhase, 2018]

Our results

Algorithms:

(New) $1/2$-approximation

(New) PTAS with resource augmentation

Lower bounds:

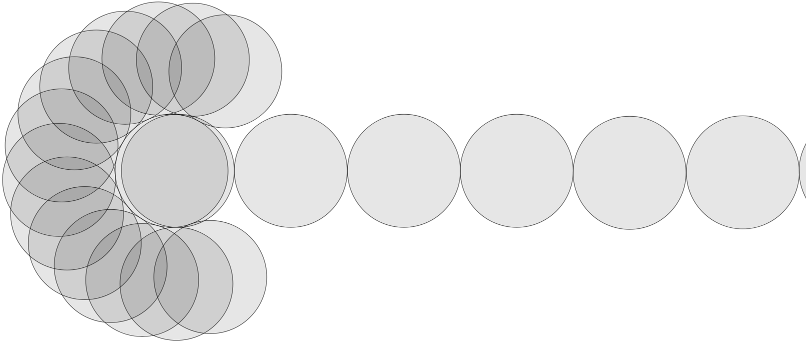
(New) NP-hardness

(New) APX-hardness with unit-area-triangles

Approximation algorithm

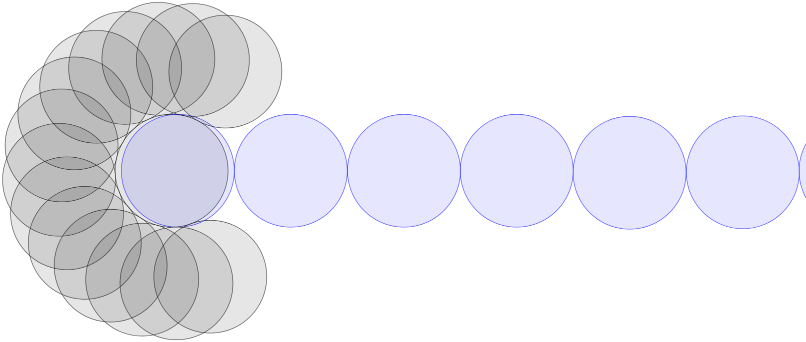
A bad Algorithm: The 1-by-1 Greedy algorithm

- $S = \{\text{an arbitrary disk}\}$
- While $|S| < k$, add one disk in S that maximizes the marginal area covered while maintaining S connected.



A bad Algorithm: The 1-by-1 Greedy algorithm

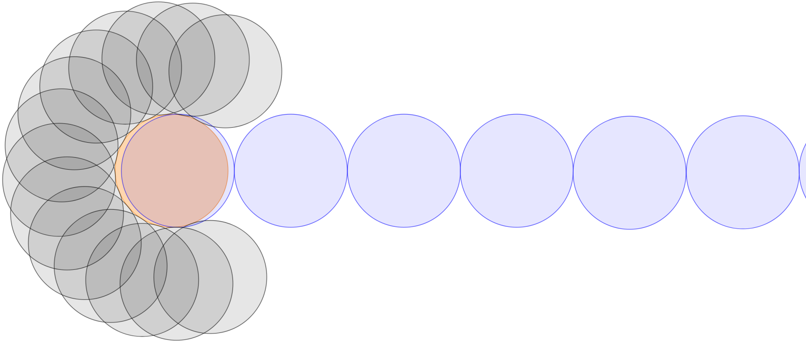
- $S = \{\text{an arbitrary disk}\}$
- While $|S| < k$, add one disk in S that maximizes the marginal area covered while maintaining S connected.



OPT$= k$

A bad Algorithm: The 1-by-1 Greedy algorithm

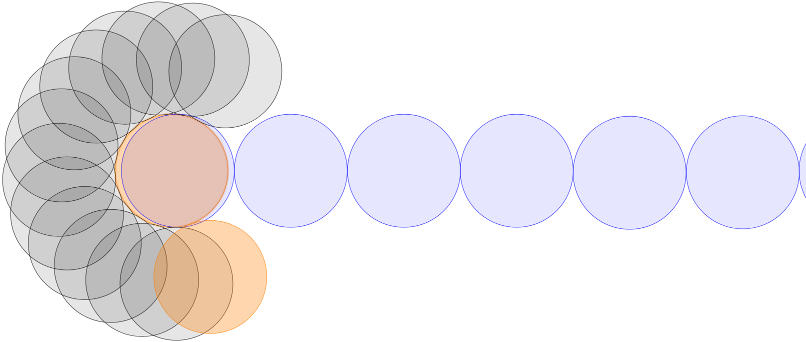
- $S = \{$an arbitrary disk$\}$
- While $|S| < k$, add one disk in S that maximizes the marginal area covered while maintaining S connected.



OPT$= k$

A bad Algorithm: The 1-by-1 Greedy algorithm

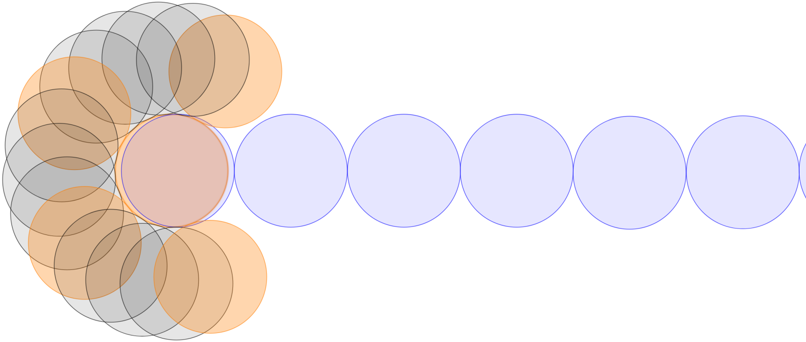
- $S = \{\text{an arbitrary disk}\}$
- While $|S| < k$, add one disk in S that maximizes the marginal area covered while maintaining S connected.



OPT$= k$

A bad Algorithm: The 1-by-1 Greedy algorithm

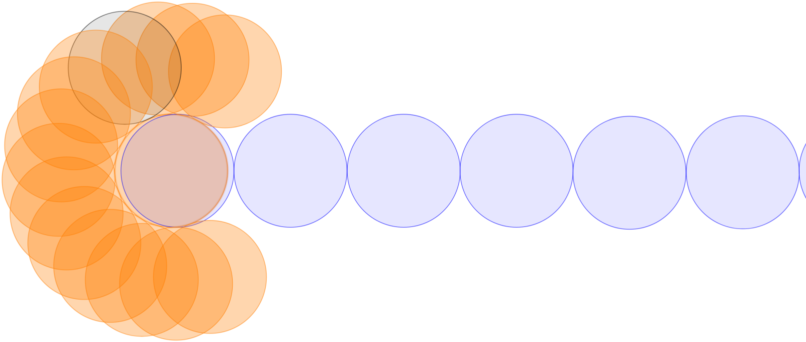
- $S = \{\text{an arbitrary disk}\}$
- While $|S| < k$, add one disk in S that maximizes the marginal area covered while maintaining S connected.



OPT$= k$

A bad Algorithm: The 1-by-1 Greedy algorithm

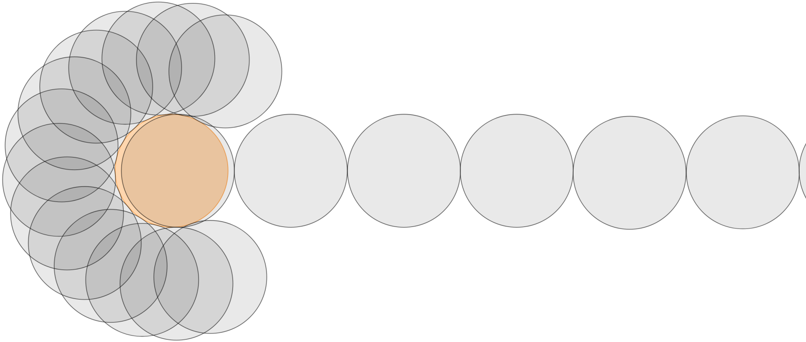
- $S = \{$an arbitrary disk$\}$
- While $|S| < k$, add one disk in S that maximizes the marginal area covered while maintaining S connected.



OPT$= k$ and 1-by-1 Greedy $\leq 9 \longrightarrow$ gap $= \Omega(k)$

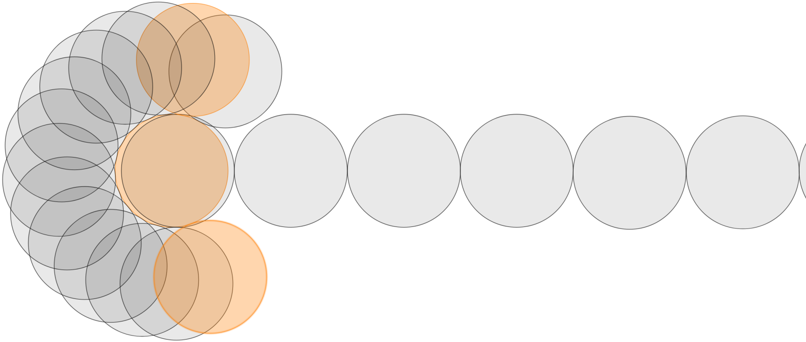
The 2-by-2 Greedy algorithm

- $S = \{$an arbitrary disk$\}$
- While $|S| < k - 1$, add two disks in S that maximize the marginal area covered while maintaining S connected.



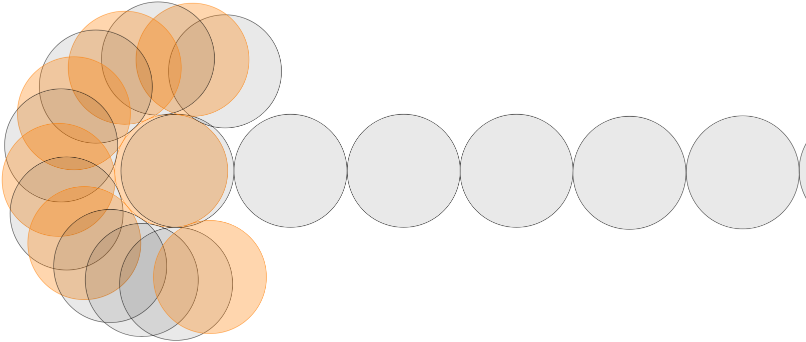
The 2-by-2 Greedy algorithm

- $S = \{\text{an arbitrary disk}\}$
- While $|S| < k - 1$, add two disks in S that maximize the marginal area covered while maintaining S connected.



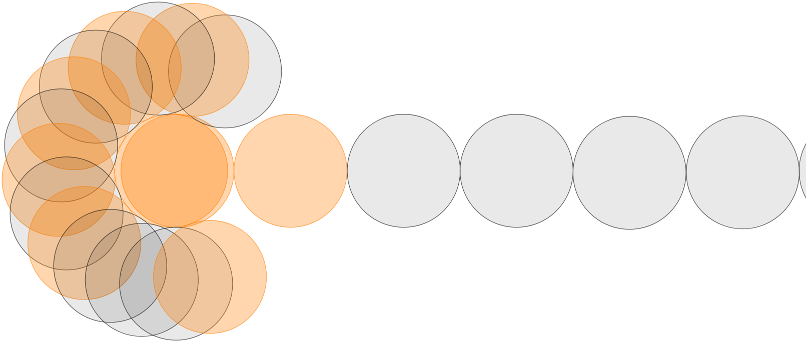
The 2-by-2 Greedy algorithm

- $S = \{$an arbitrary disk$\}$
- While $|S| < k - 1$, add two disks in S that maximize the marginal area covered while maintaining S connected.



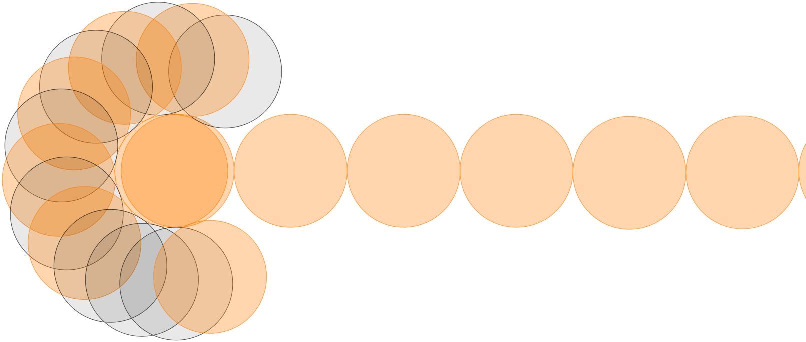
The 2-by-2 Greedy algorithm

- $S = \{\text{an arbitrary disk}\}$
- While $|S| < k - 1$, add two disks in S that maximize the marginal area covered while maintaining S connected.



The 2-by-2 Greedy algorithm

- $S = \{$an arbitrary disk$\}$
- While $|S| < k - 1$, add two disks in S that maximize the marginal area covered while maintaining S connected.



The 2-by-2 Greedy algorithm

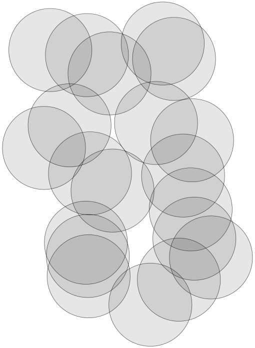
- $S = \{$an arbitrary disk$\}$
- While $|S| < k - 1$, add two disks in S that maximize the marginal area covered while maintaining S connected.

Theorem (NEW): The 2-by-2 Greedy algorithm gives a $\frac{1}{2}$-approximation of connected unit-disk k-coverage problem, and it is tight.

Proof sketch

First phase

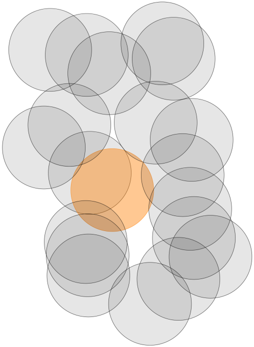
S is not a dominating set



Proof sketch

First phase

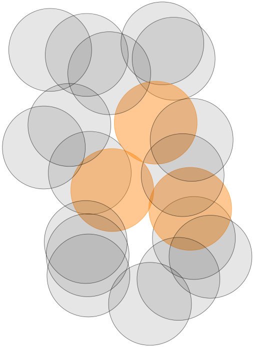
S is not a dominating set



Proof sketch

First phase

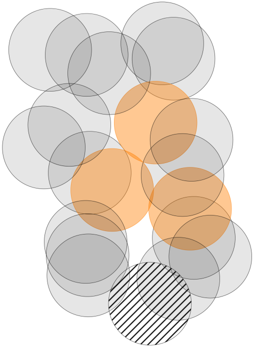
S is not a dominating set



Proof sketch

First phase

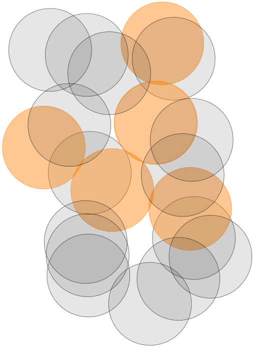
S is not a dominating set



Proof sketch

First phase

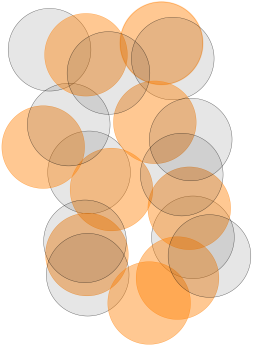
S is not a dominating set



Proof sketch

First phase

S is not a dominating set

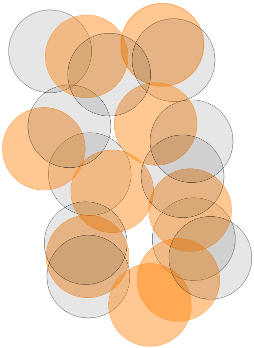


$\text{area}(S) \geq |S|/2$

Proof sketch

First phase

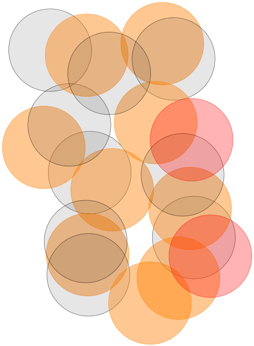
S is not a dominating set



$\text{area}(S) \geq |S|/2$

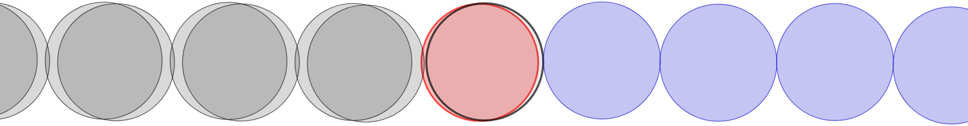
Second phase

connectivity is guaranteed



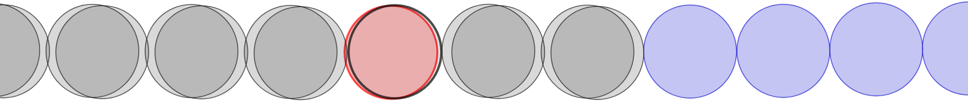
use monotone submodularity.

Theorem: The 2-by-2 Greedy algorithm gives a $\frac{1}{2}$-approximation of connected unit-disk k-coverage problem, **and it is tight**.



Going above $1/2$?

a t-by-t Greedy algorithm, with $t \geq 3$? **No.**

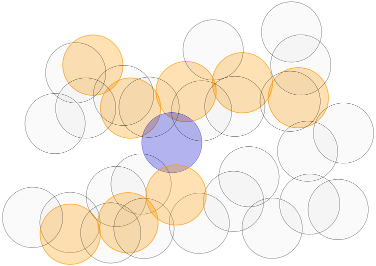


Theorem (NEW): PTAS with resource augmentation

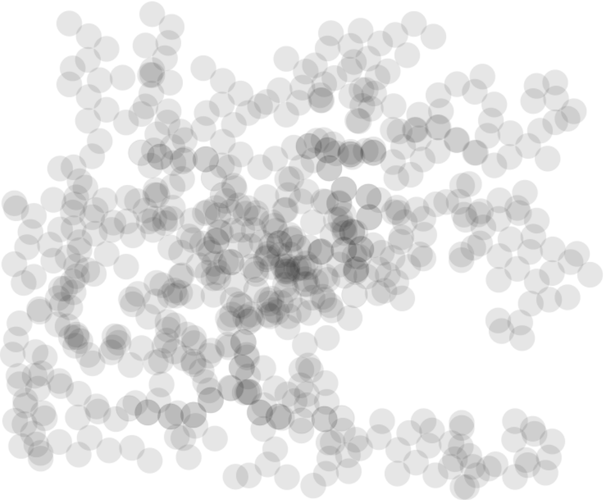
We can find in time $n^{O(1/\varepsilon)}$

- a set S of k input disks, such that area$(S) \geq (1-\varepsilon)OPT(k)$
- a set S_{add} of at most εk additional disks such that $S \cup S_{add}$ is connected.

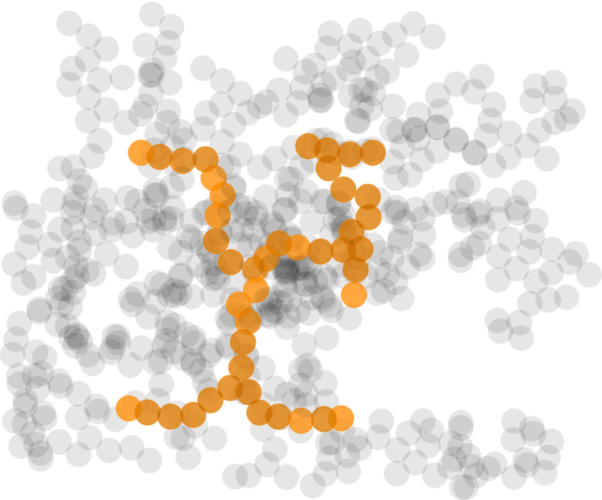
Algorithm: Shifted quadtree (Arora), m-guillotine subdivision (Mitchell)



Proof with Arora's framework

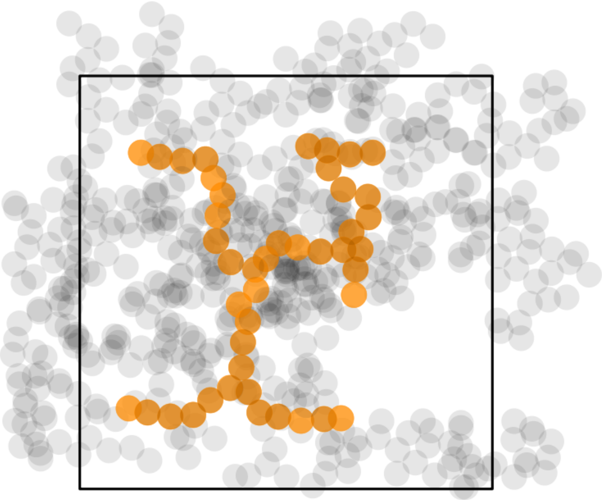


Proof with Arora's framework



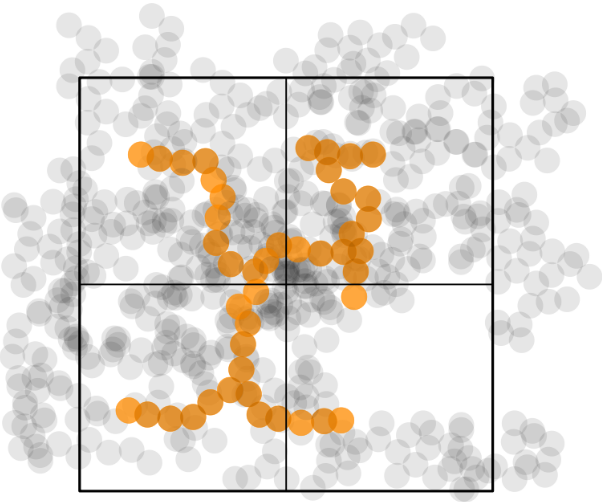
OPT

Proof with Arora's framework



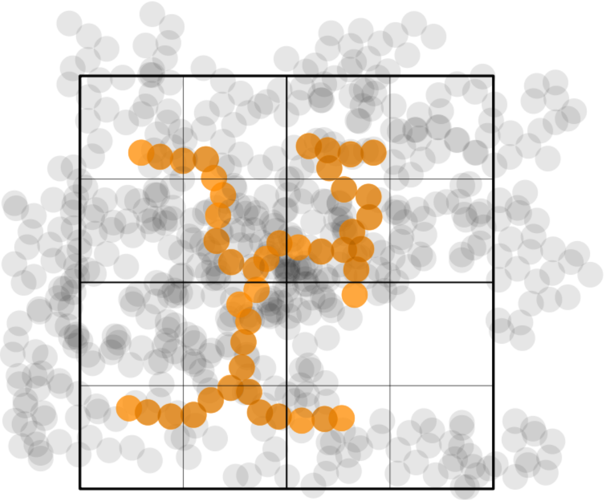
OPT

Proof with Arora's framework



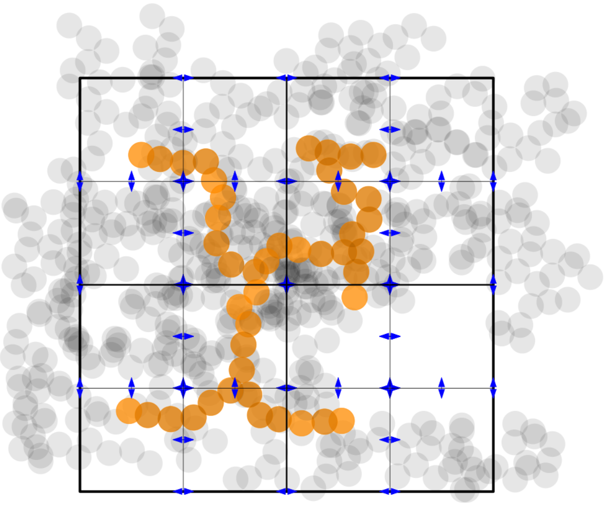
OPT

Proof with Arora's framework



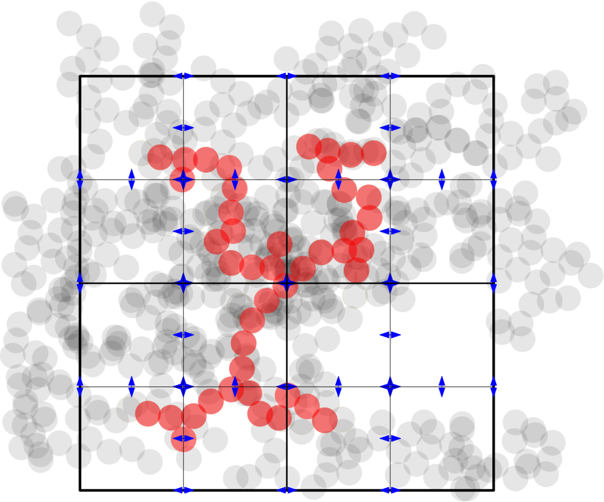
OPT

Proof with Arora's framework



OPT

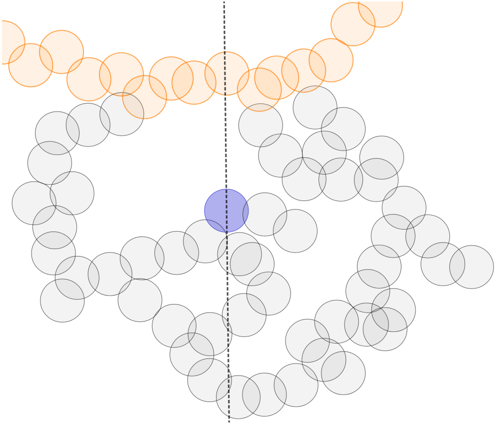
Proof with Arora's framework



OPT $\longrightarrow \exists$ portal-respecting near-optimal solution ??

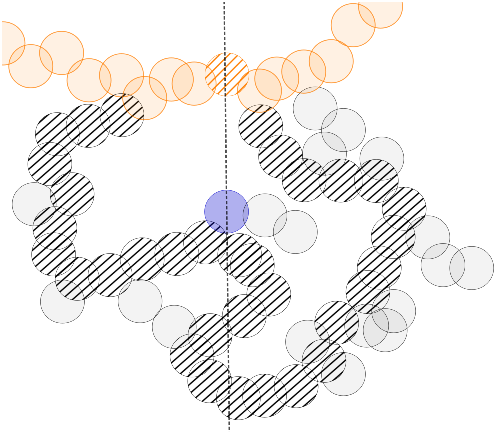
Proof with Arora's framework

Can we make short detours ?



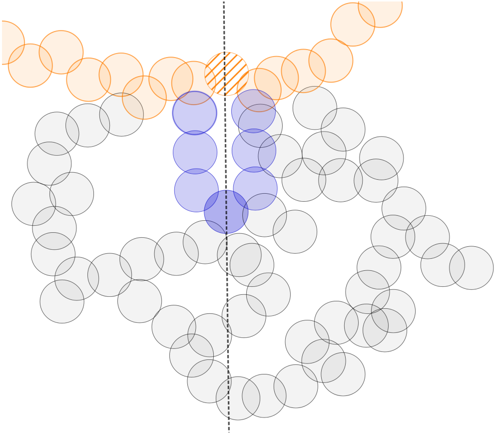
Proof with Arora's framework

Can we make short detours ?



Proof with Arora’s framework

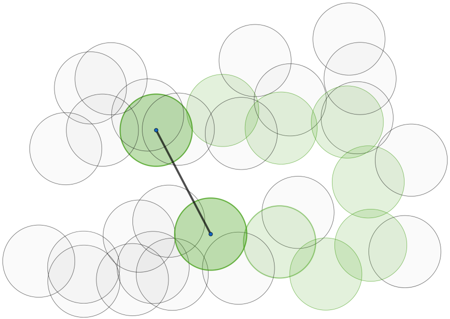
Can we make short detours ? Yes if we allow few additional disks



PTAS under resource augmentation

We can find in time $n^{O(1/\varepsilon)}$ a k-set $S \subseteq D$ and a set S_{add} of at most εk additional disks such that

1. $S \cup S_{add}$ is connected,
2. $\text{area}(S) \geq (1 - \varepsilon)OPT$



corollary

$\exists$ PTAS when

distance in intersection graph $=$ O(Euclidean distance)

Our results:

- $1/2$-approximation
- PTAS with resource augmentation
- NP-hardness
- APX-hardness with unit-area-triangles.

$$\Downarrow$$

$\exists$ PTAS for connected unit-disk k-coverage?