Maximizing Covered Area in the Euclidean Plane with Connectivity Constraint

Chien-Chung Huang  Mathieu Mari\textsuperscript{1}  Claire Mathieu  Joseph S. B. Mitchell  Nabil H. Mustafa

\textsuperscript{1}École Normale Supérieure, Université PSL, Paris
Connected Unit-disk $k$-coverage Problem

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

**Output:** A connected subset $S$ of size $k$

**Goal:** Maximize the area covered by the union of disks in $S$
Connected Unit-disk $k$-coverage Problem

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

**Output:** A connected subset $S$ of size $k$

**Goal:** Maximize the area covered by the union of disks in $S$
Connected Unit-disk $k$-coverage Problem

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

**Output:** A connected subset $S$ of size $k$

**Goal:** Maximize the area covered by the union of disks in $S$
Generalisations

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

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

Related results

**$k$-coverage:** optimal greedy $1 - \frac{1}{e}$ 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:

• $1/2$-approximation algorithm
• PTAS with resource augmentation

Lower bounds:

• NP-hardness
• APX-hardness with unit-area-triangles
Approximation algorithm
First try: 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.
First try: 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.

$\text{OPT} = k$
**First try: 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.

\[ \text{OPT} = k \]
First try: 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.

\[ \text{OPT} = k \]
First try: 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.

$OPT = k$
First try: 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.

\( \text{OPT} = k \) and 1-by-1 Greedy \( \leq 9 \) \( \rightarrow \) gap = \( \Omega(k) \)
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

- $S = \{\text{an arbitrary disk}\}$
- 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

- $S = \{\text{an arbitrary disk}\}$
- 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**

- $S = \{\text{an arbitrary disk}\}$
- 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**

- \( S = \{ \text{an arbitrary disk} \} \)
- 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

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

Theorem: 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
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

\[
\text{area}(S) \geq \frac{|S|}{2}
\]
Proof sketch

First phase

$S$ is not a dominating set

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.
Improving 1/2?
a $t$-by-$t$ Greedy algorithm, with $t \geq 3$? No.
Theorem: PTAS with resource augmentation

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

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

Algorithms: Shifted quadtree/ $m$-guillotine subdivision
Proof with Shifted Quadtree framework

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

Can we make short detours ?

Yes if we allow few additional disks /one.pnum/two.pnum
Proof with Shifted Quadtree framework

OPT

−→∃ portal-respecting near-optimal solution??

Can we make short detours?

Yes if we allow few additional disks
Proof with Shifted Quadtree framework

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

OPT
Proof with Shifted Quadtree framework

OPT

−→∃ portal-respecting near-optimal solution ??

Can we make short detours?

Yes if we allow few additional disks

one.pnum/two.pnum
Proof with Shifted Quadtree framework

OPT → ∃ portal-respecting near-optimal solution ??

Can we make short detours? Yes if we allow few additional disks / one.pnum/two.pnum
Proof with Shifted Quadtree framework

OPT
Proof with Shifted Quadtree framework

$\text{OPT} \rightarrow \exists$ portal-respecting near-optimal solution ??
Can we make short detours?
Can we make short detours?
Can we make short detours? Yes if we allow few additional disks.
Theorem: PTAS with resource augmentation

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

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

Corollary

$\exists$ PTAS when

\[
\text{distance in intersection graph} = O(\text{Euclidean distance})
\]
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?