APPROXIMATION ALGORITHMS
for
SPREADING POINTS

SERGIO CABELLO
IMFM, LJUBLJANA, SLOVENIA

research done at Utrecht University

THE ABSTRACT PROBLEM

- (X, d) metric space

  $S_1, \dots, S_n \subseteq X$

- t-distant representatives: [Fiala et. al '02]

  $$\begin{array}{ccc} x_1, \dots, & x_n & \\ \cap & \cap & \text{s.t. } d(x_i, x_j) \geq t \quad \forall i \neq j \\ S_1, \dots, & S_n & \end{array}$$

- Optimization problem:

  maximize t that admits t-representatives

OUR PROBLEMS

- $(\mathbb{R}^2, L_\infty)$

  $S_1, \dots, S_n$ disks (squares in L_∞)

  Choose $p_i \in S_i$ maximizing the distance of the closest pair.

- $(\mathbb{R}^2, L_2)$

  $S_1, \dots, S_n$ (congruent) disks

  Choose $p_i \in S_i$ maximizing the distance of c.p.

PREVIOUS/RELATED WORK

- Fiala et. al
- Baur & Fekete

NP-hard to get PTAS

- Baur & Fekete: Choose K points inside a polygonal region maximizing the distance of c.p.
  NP-hard to get a PTAS. $(3/2)$-approximation in L_∞
- Packing problems – Map labelling problems.

OUR RESULTS

SPACE	REGIONS	RESULT
$(\mathbb{R}^2, L_\infty)$	disks	2-approximation, $O(n\sqrt{n}\log^2 n)$ *
$(\mathbb{R}^2, L_2)$	disks	$(8/3)$-approximation, $O(n^2)$
	congruent disks	(~ 2.23)-approximation, $O(n^2)$

* today's aim

OUTLINE

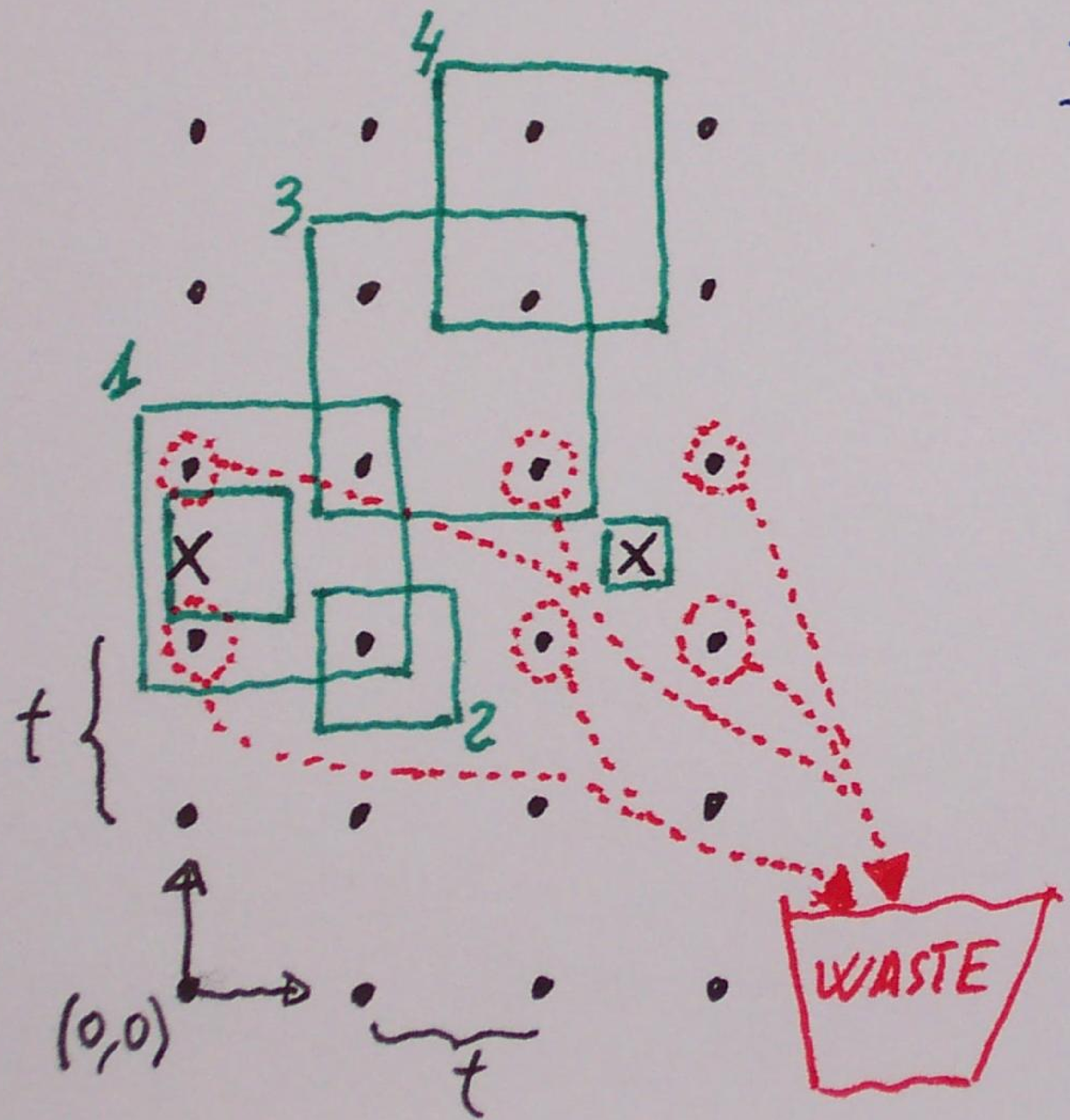
- ✓ The abstract problem / Our problem
- ✓ Previous/Related work
- ✓ Our results
- Approximate placement algorithm
- Efficiency of approximate placement
- Decision $\rightsquigarrow$ Optimization
- Comments on L_2 case

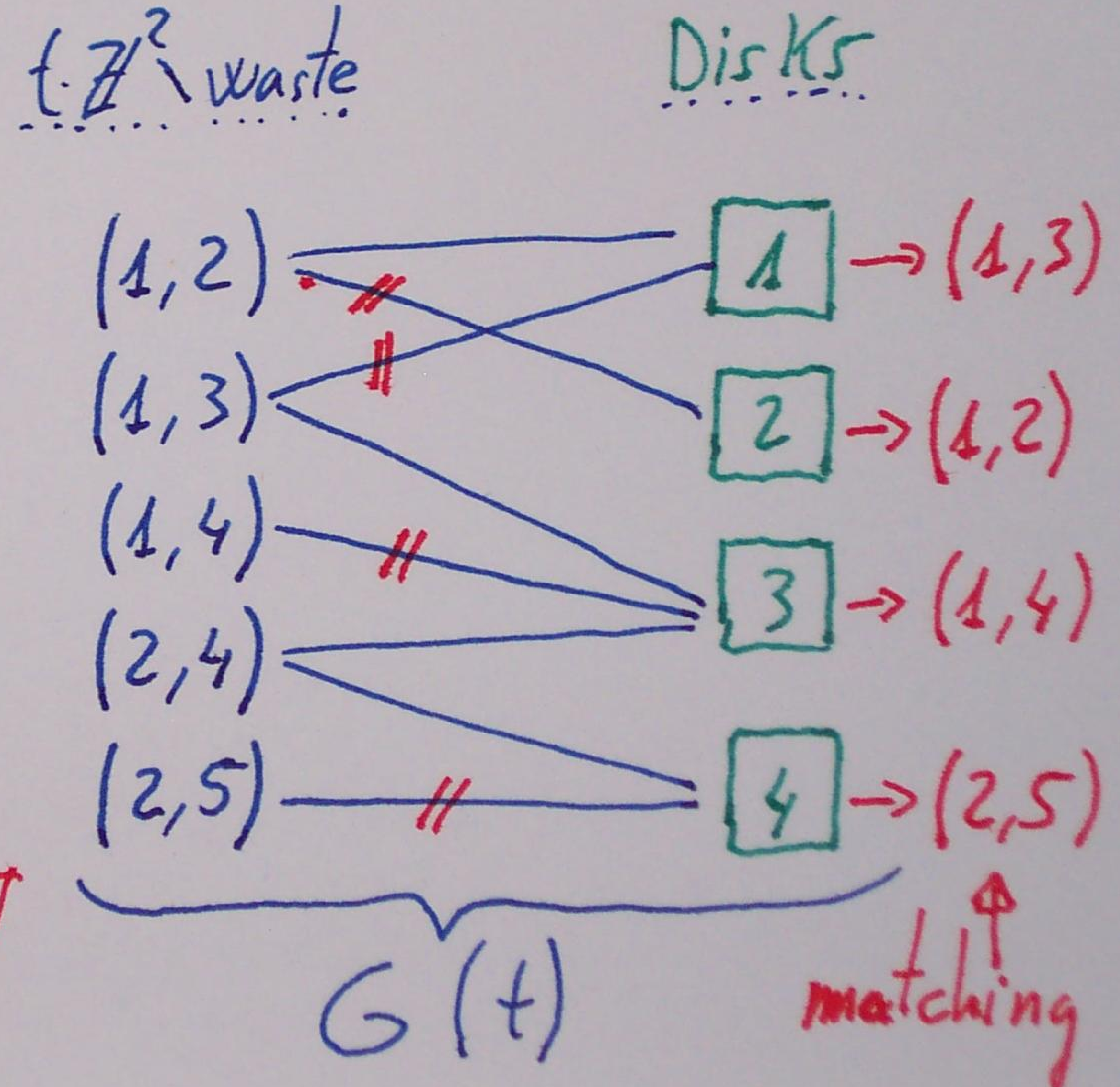
L_∞

2-approximation

PLACEMENT ALGORITHM (t)

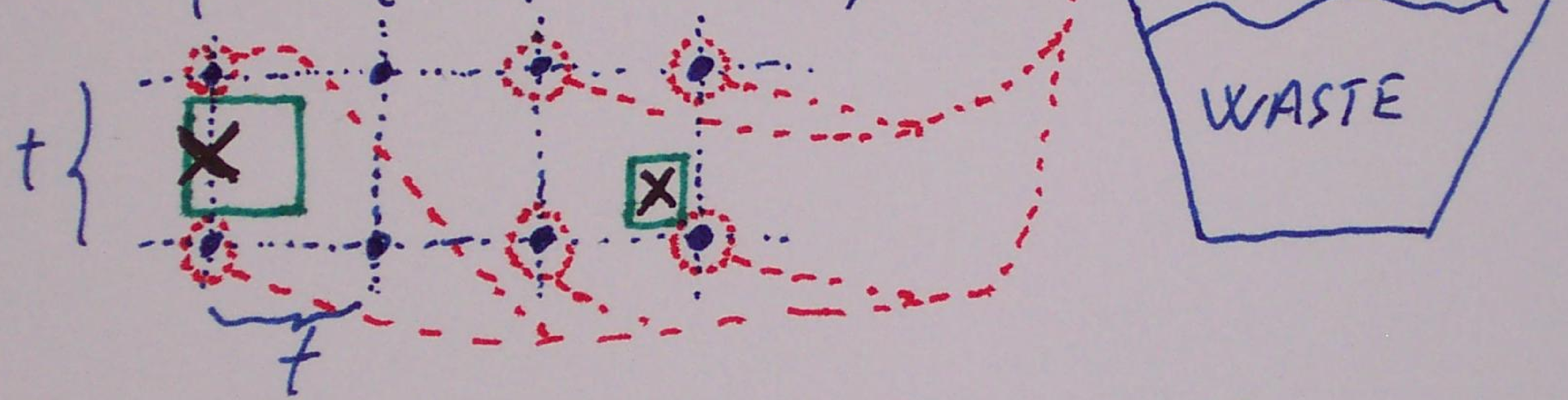
<u>Idea</u>: Try to place points at $t \cdot \mathbb{Z}^2$.





Placement Algorithm (t)

- $\text{WASTE} := \emptyset$;
- if $S_i \cap (t\cdot\mathbb{Z}^2) = \emptyset$, two cases



- construct bipartite $G(t) = \left(\{S_1,\dots,S_n\} \,\dot\cup\, (t\cdot\mathbb{Z}^2 \setminus \text{WASTE}),\ \{(p,S_i) \mid p\in S_i\}\right)$,
- matching in $G(t)$

  ↳ Placement.

PLACEMENT ALGORITHM (t)

Let t^* be the optimal solution.

Lemma: If $2t \leq t^*$ then:

- $G(t)$ has a matching ⇝ we get placement;
- c.p. of placement is t apart.

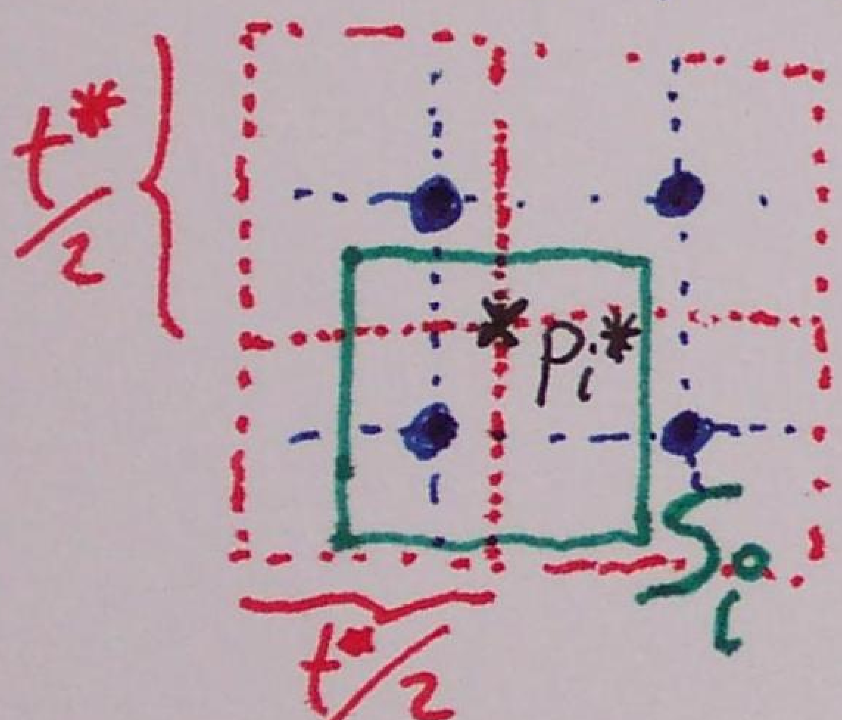
We may have a huge graph $G(t)$. !

↳ Modify algorithm

PLACEMENT ALGORITHM (t)

<u>proof</u>: $p_1^*, \dots, p_n^*$ attain optimum t^*

$\Rightarrow B(p_1^*, t^*/2), \dots, B(p_n^*, t^*/2)$ disjoint.



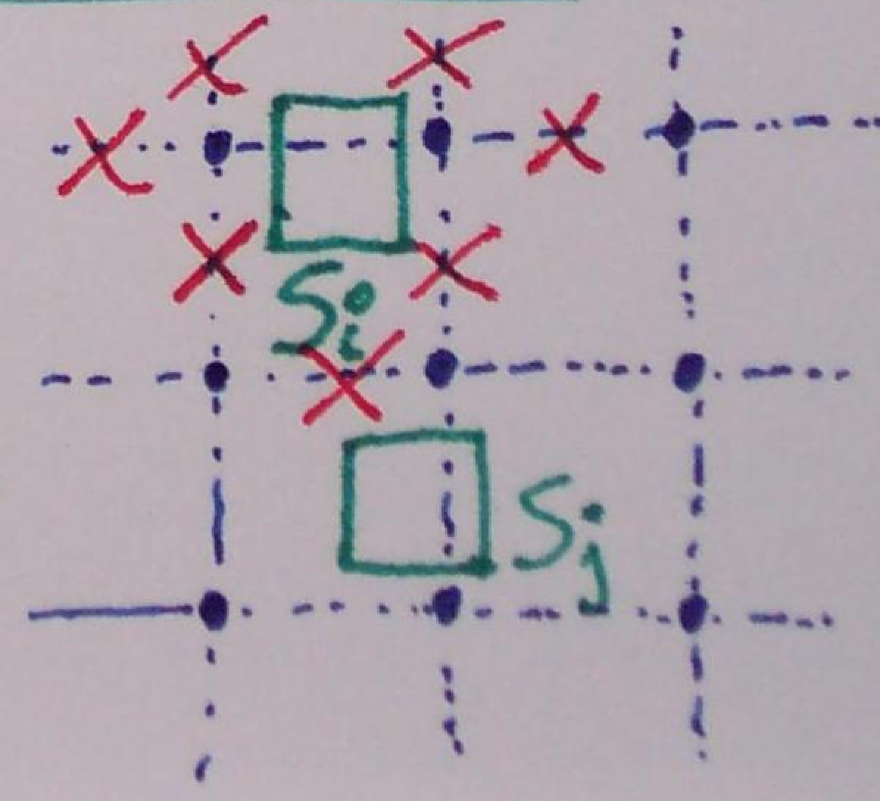
$\Rightarrow$ If $t \le t^*/2$, $G(t)$ has an edge contained in $B(p_i^*, t^*/2) \Rightarrow \exists$ matching

---

- p_i, p_j from PLACEMENT(t)

If $p_i, p_j \in t \cdot \mathbb{Z}^2$, no problem.

Otherwise



$\boxtimes$

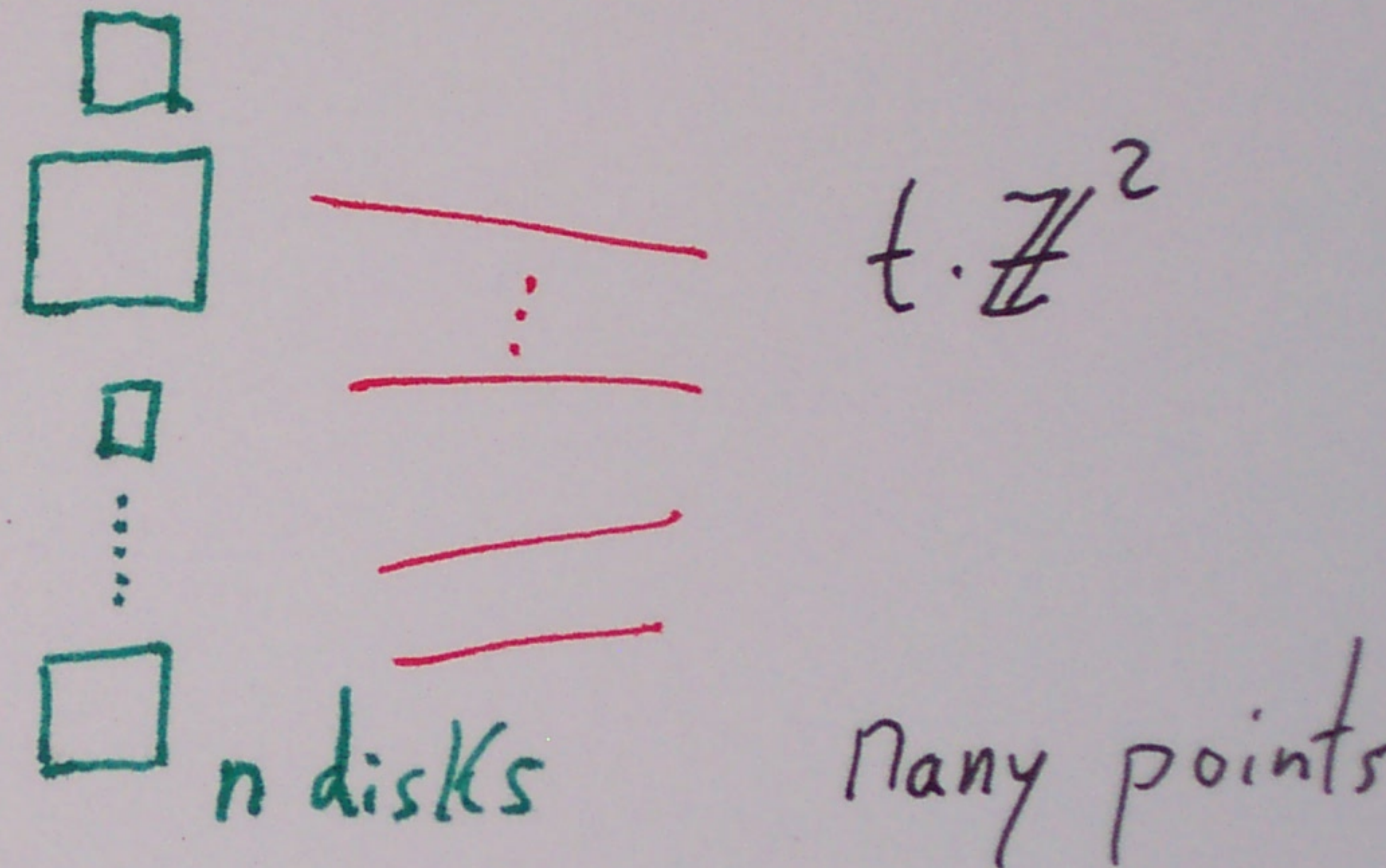
OUTLINE

- ✓ The abstract problem / Our problem
- ✓ Previous/Related work
- ✓ Our results
- ✓ Approximate placement algorithm
- Efficiency of approximate placement
- Decision vs Optimization
- Comments on L_2 case

L_∞
2-approximation

EFFICIENCY PLACEMENT (t)

$G(t)$ looks like



$\rightsquigarrow$ Each disk needs degree $\leq n$

$\rightsquigarrow$ For each disk, take $5n$ points $(|\text{WASTE}| \leq 4n)$

$\rightsquigarrow$ $G(t)$ has $\boxed{n} + O(n^2)$ vertices.

Lemma: PLACEMENT $\in$ P. $\boxtimes$

EFFICIENCY PLACEMENT (t)

Lemma: PLACEMENT (t) can be done in $O(n\sqrt{n}\log n)$ time.

proof: Two steps:

- considering $O(n\sqrt{n})$ points in total (1)
- geometry helps for matching (2)

  T disks, P points $\rightsquigarrow O(P \log P + \sqrt{T} \cdot T \cdot \log P)$

$\rightsquigarrow O(n\sqrt{n}\log n)$ time ⊠

EFFICIENCY PLACEMENT (t)

- considering $O(n\sqrt{n})$ points in total (1)

shrink/grow disks to have $\sim 5n$ points (radius $\sim t\sqrt{n}$)

disks intersecting

$\geq \sqrt{n}$ disks

- compute $\bigcup$ disks
- area$(\bigcup) \leq \sqrt{n} \cdot O(t\sqrt{n})^2$
  $\approx t^2 n\sqrt{n}$
- report $t \cdot \mathbb{Z}^2 \cap \bigcup$

$< \sqrt{n}$ disks

- compute WASTE
- for each disk S_i, select $\sqrt{n}$ points in $S_i \setminus$ WASTE
- each $p \in$ WASTE troubles $\leq \sqrt{n}$ times

$O(n\sqrt{n})$ points in $O(n\sqrt{n}\log n)$ time

EFFICIENCY PLACEMENT (t)

- Geometry helps for matching [Efrat etal.] (2)

T disks, P points, $G \equiv (T \sqcup P, \{\{t, p\} \mid p \in t\})$

Use semi-dynamic D.S. for P $\begin{cases} \text{construction } O(P \log P) \\ \text{delete point } O(\log P) \\ \text{report point in } t \ O(\log P) \end{cases}$

Use standard blocking flows $\rightarrow$ $O(\log P)$ to find an edge

$\rightsquigarrow O(\sqrt{T})$ phases $\times$ $O(T \log P)$ per phase $\equiv O(T\sqrt{T} \log P)$

OUTLINE

- ✓ The abstract problem / Our problem
- ✓ Previous/Related work
- ✓ Our results
- ✓ Approximate placement algorithm
- ✓ Efficiency of approximate placement
- Decision $\rightsquigarrow$ Optimization

L_∞ 2-approximation

- Comments on L_2 case

DECISION $\rightsquigarrow$ OPTIMIZATION

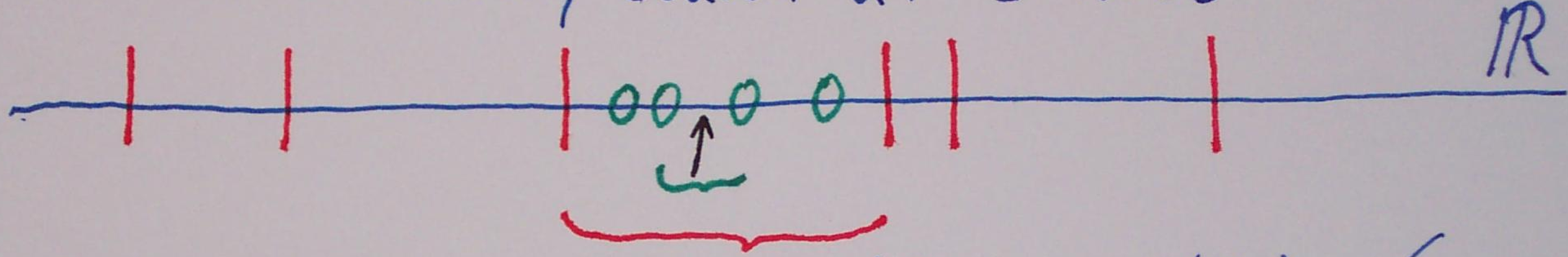
Lemma: If $2t \leq t^*$, PLACEMENT(t) ✓

Corollaries: How to prove PLACEMENT(t) is 2-approxi

- If PLACEMENT(t) ✓ and PLACEMENT(t) ✗ after translation $\rightsquigarrow$ $t > t^*/2$
- If $t > t'$, PLACEMENT(t) ✓, PLACEMENT(t') ✗ $\rightsquigarrow$ $t > t^*/2$
- If PLACEMENT(t) ✓ and PLACEMENT$(t+\varepsilon)$ ✗ for infinitessimal $\varepsilon > 0$ $\rightsquigarrow$ $t \geq t^*/2$

DECISION $\rightsquigarrow$ OPTIMIZATION

Main idea: binary search at 3 levels



Invariant: $t_1 < t_2$ s.t. $\begin{cases} \text{PLACEMENT}(t_1) \ \checkmark \\ \text{PLACEMENT}(t_2) \ \times \end{cases}$

Objective: $t_1 < t_2$ like above and

$$G(t_1 + \varepsilon) \cong G(t_2)$$

$\Longrightarrow$ t_1 is a 2-approximation

DECISION → OPTIMIZATION

first level → finding which disks are huge
↳ shrink them

binary search in

$$\left\{ \frac{r_1}{3\sqrt{n}}, \dots, \frac{r_n}{3\sqrt{n}}, 4 r_n \right\} \to O(\log n) \text{ steps}$$

each S_i in $G\left(\frac{r_1}{3\sqrt{n}}\right)$ has degree $\geq n$

$\Rightarrow \exists$ matching, $t^* \geq \frac{r_1}{3\sqrt{n}}$

If PLACEMENT$(4r_n)$ ✓

$\Rightarrow$ Disks are disjoint

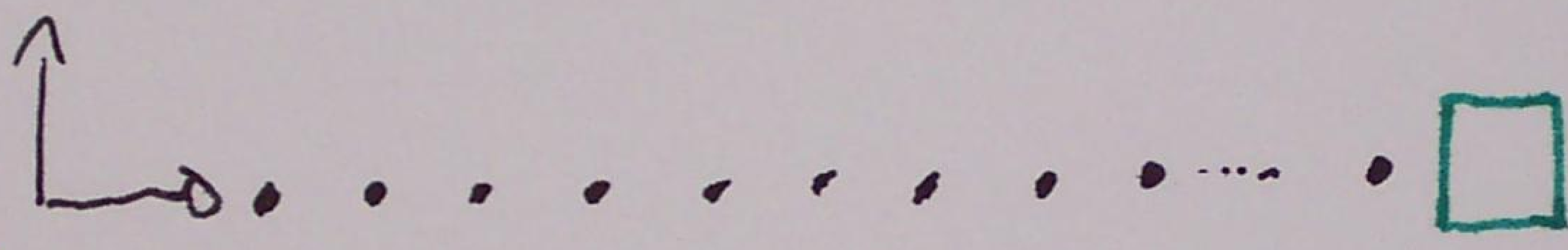
$\Rightarrow$ easy a 2-approximation

$\approx \Rightarrow t_1 = \frac{r_i}{3\sqrt{n}} < 4 r_i = t_2$; Shrink to radius $O(r_i \sqrt{n})$

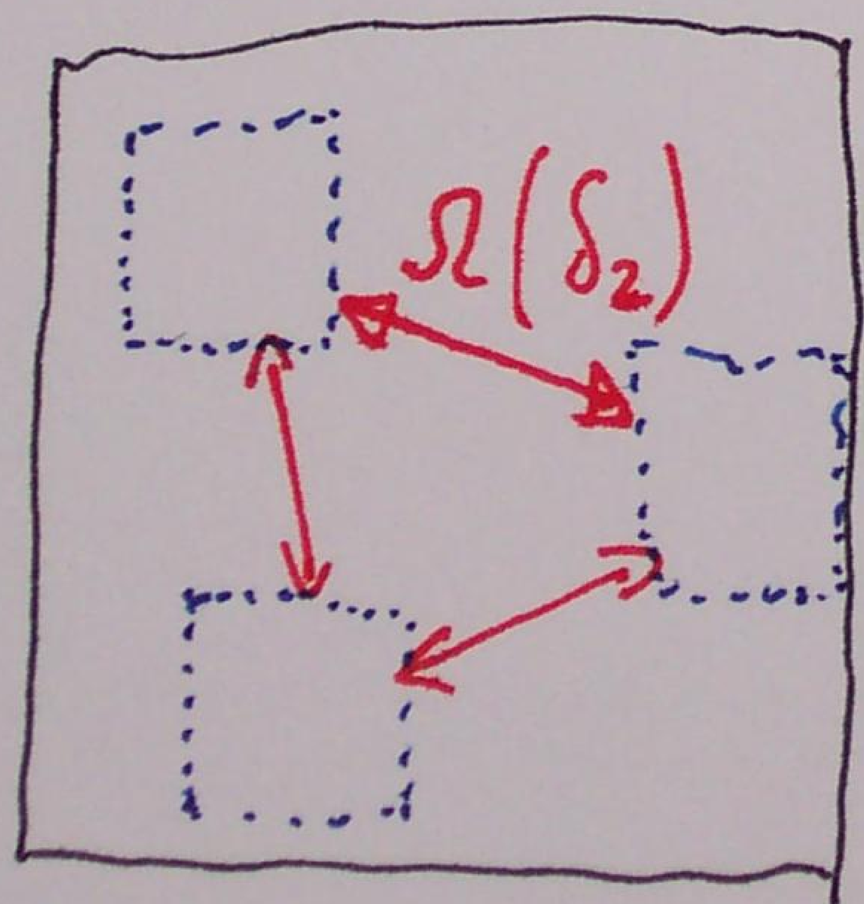
DECISION $\rightsquigarrow$ OPTIMIZATION

Decompose the problem into clusters.

Reason: things should be around the origin



otherwise, $G(t)$ changes many times.

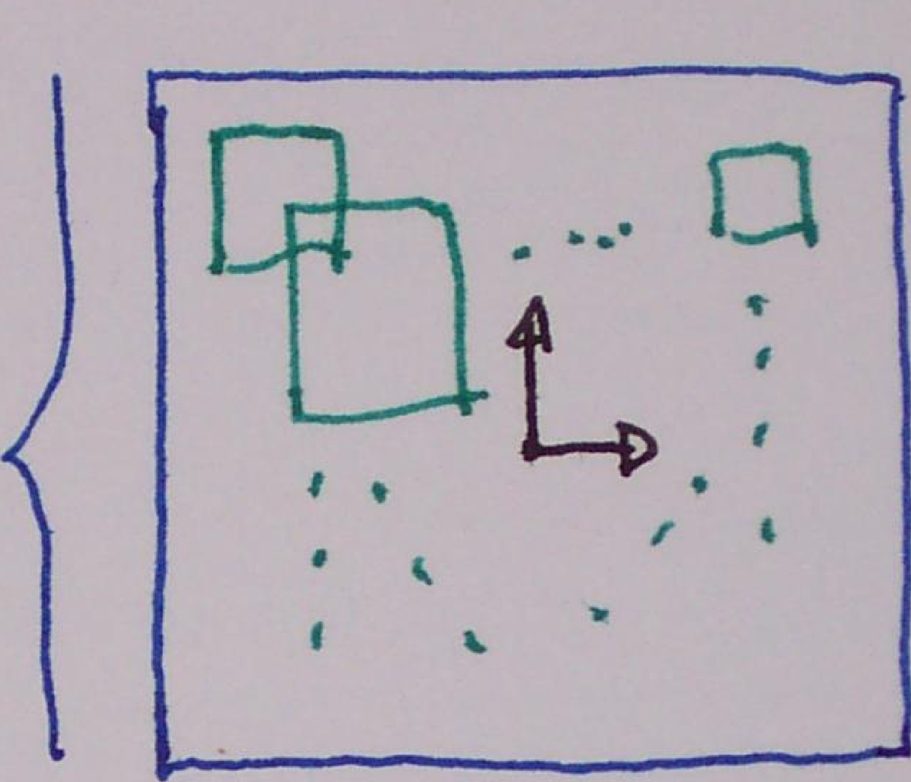


bounding box
original problem

DECISION ⇝ OPTIMIZATION

second level

$n \cdot t_2 \geq$



bounding box B

Binary search in values t that $t \cdot \mathbb{Z}^2 \cap B$ changes

→ $O(\log n)$ steps because $\sqrt{n} \cdot t_1 \simeq t_2$

$\Rightarrow$ $t_1 < t_2$ PLACEMENT(t_1) ✓ PLACEMENT(t_2) ✗

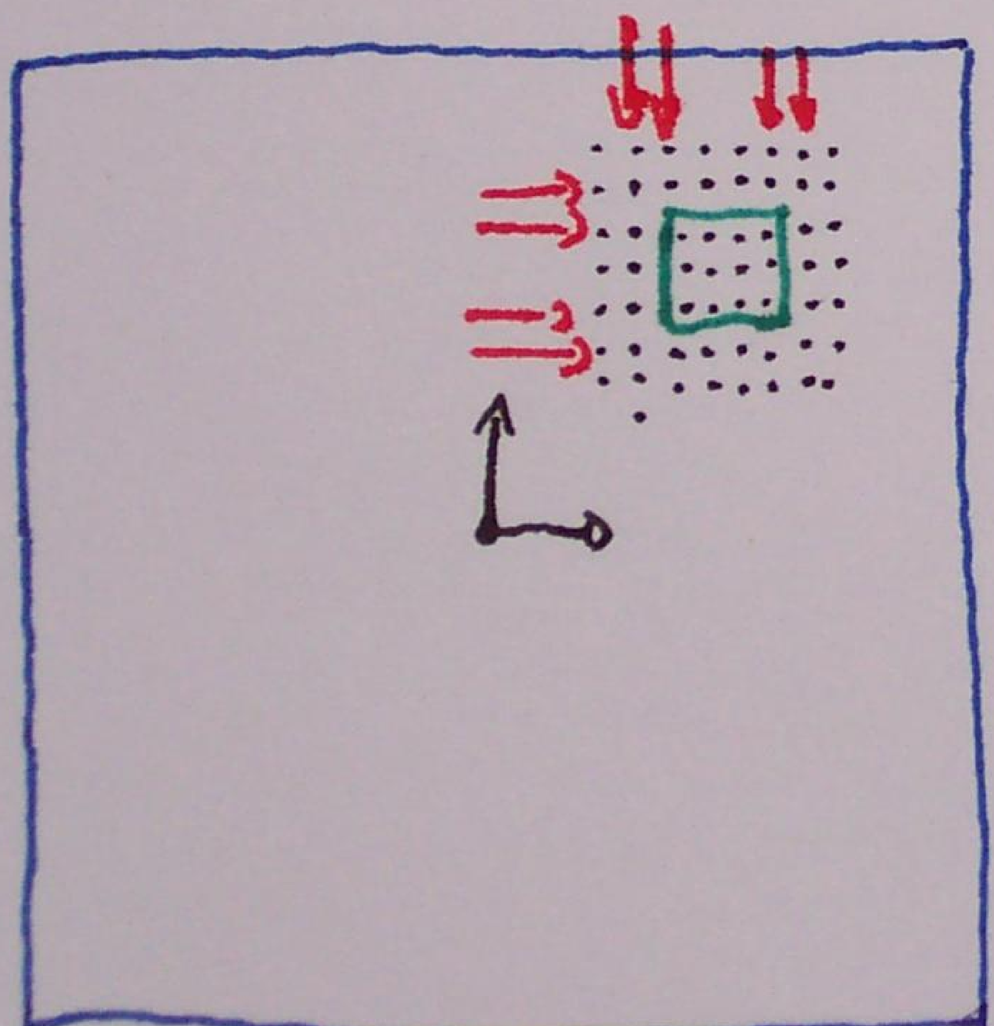
$(\varepsilon + t_1) \cdot \mathbb{Z}^2 \cap B \cong t_2 \cdot \mathbb{Z}^2 \cap B$

DECISION $\rightsquigarrow$ OPTIMIZATION

third level

for each level there is at most one $t_i' \in [t_1, t_2)$

s.t. $\quad t_i' \cdot \mathbb{Z}^2 \cap S_i \neq (t_i' + \varepsilon) \cdot \mathbb{Z}^2 \cap S_i$



(because changes in ∂B are quicker)

$\rightsquigarrow O(\log n)$ binary steps

in $\{t_1', \dots, t_n'\}$

B

DECISION $\rightsquigarrow$ OPTIMIZATION

Theorem: In $(\mathbb{R}^2, L_\infty)$, regions are disks, we have a 2-approximation in $O(n\sqrt{n}\log^2 n)$.

proof: $3 \times O(\log n) \times O(n\sqrt{n}\log n)$ time

we find $t_1 < t_2$ s.t.

PLACEMENT (t_1) ✓

PLACEMENT (t_2) ✗

$$G(t_1+\varepsilon) \cong G(t_2)$$

$\boxtimes$

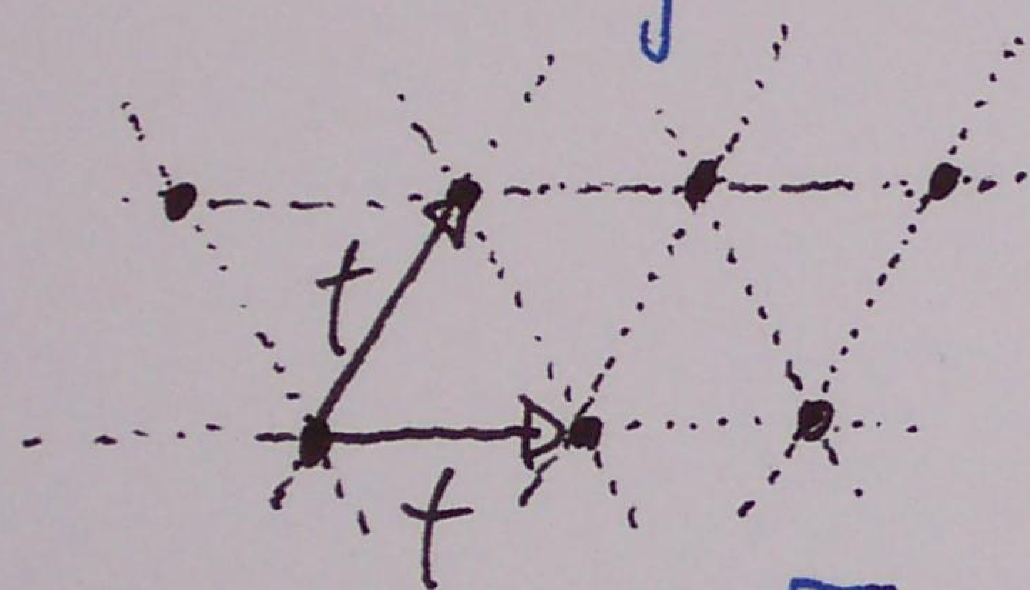
OUTLINE

- The abstract problem / Our problem
- Previous/Related work
- Our results
- Approximate placement algorithm
- Efficiency of approximate placement
- Decision $\rightsquigarrow$ Optimization
- Comments on L_2 case

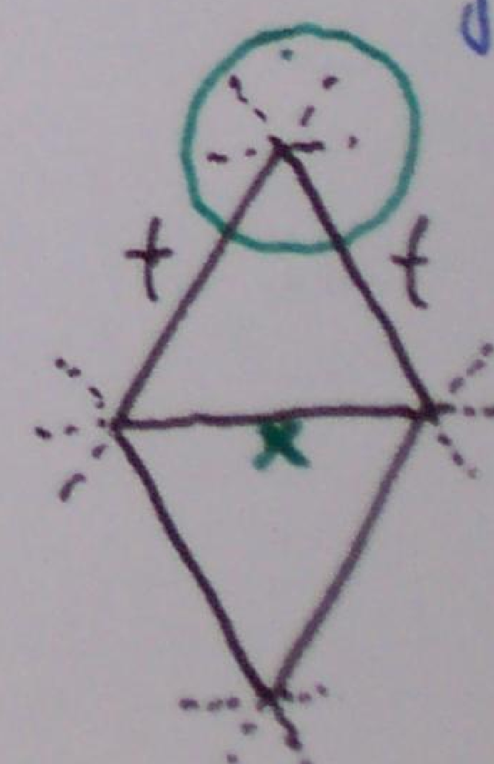
L_∞
2-approximation

L_2 CASE

- Use triangular lattice for PLACEMENT(t)



- If $t \leq \frac{\sqrt{3}}{4} t^*$ $\Rightarrow$ $G(t)$ has a matching
- C.p. in PLACEMENT(t) is $\geq \frac{\sqrt{3}}{2} t$



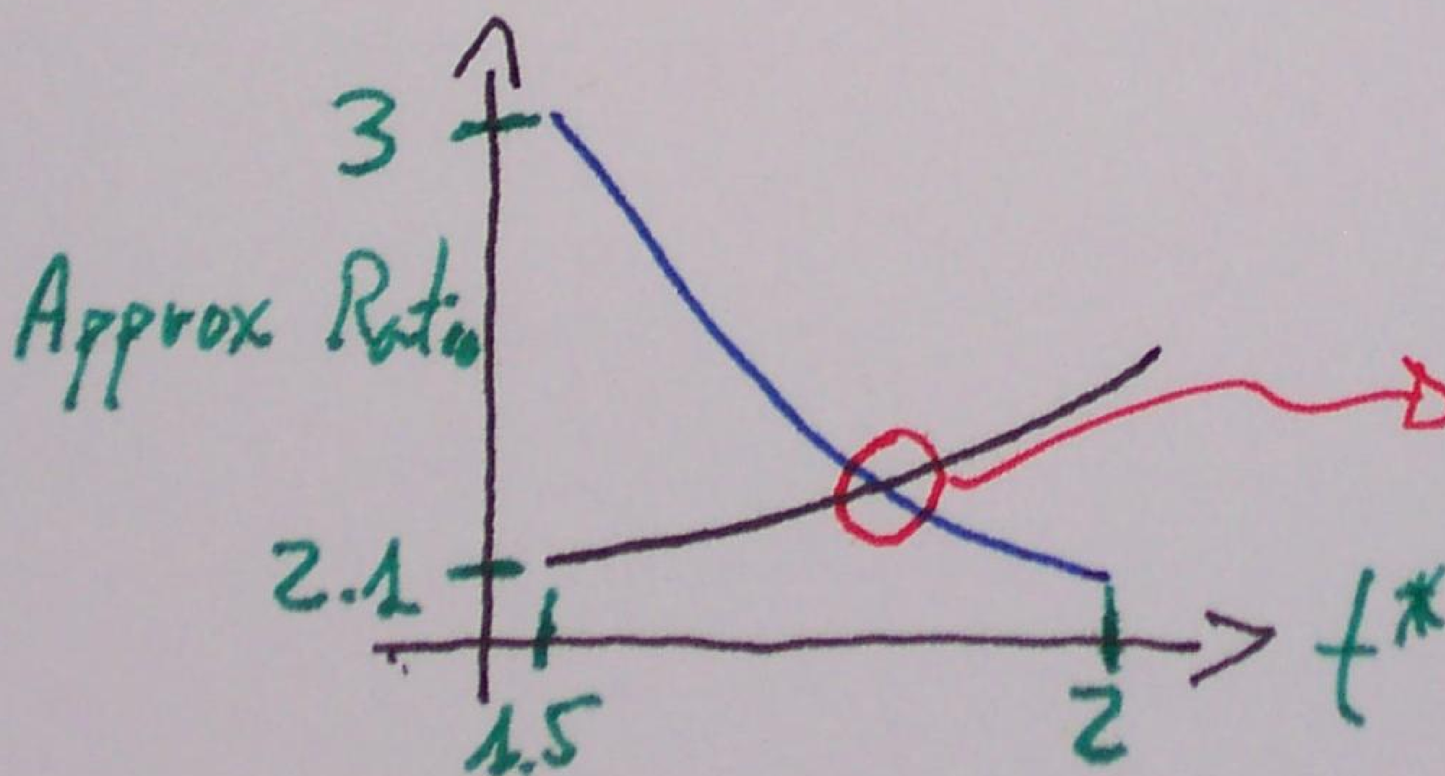
$\Rightarrow$ $8/3$ -approximation

L_2 CASE – Unit disks

- Tighter analysis of PLACEMENT shows

$$\frac{4t^*}{-\sqrt{3}+\sqrt{3}\,t^*+\sqrt{3+2t^*-t^{*2}}}\text{-approximation.}$$

- CENTERS gives a $\frac{t^*}{t^*-1}$-approximation.

- 

Taking the best → 2.23-approximation

OUTLINE

- ✓ The abstract problem / Our problem
- ✓ Previous/Related work
- ✓ Our results
- ✓ Approximate placement algorithm
- ✓ Efficiency of approximate placement
- ✓ Decision vs Optimization
- ✓ Comments on L_2 case

L_∞
2-approximation

OPEN PROBLEM

- If S_i's disjoint $\leadsto$ CENTERS is 2-approximation

Improve it !

Get a T-approximation with $T < 2$
for disjoint disks.