Efficient Approximation Schemes for Geometric Problems?

Dániel Marx

Humboldt-Universität zu Berlin

`dmarx@informatik.hu-berlin.de`

European Symposium on Algorithms

October 4, 2005

Overview

- An $n^{O(1/\epsilon^2)}$ approximation scheme is grossly inefficient for small ϵ.
- Parameterized complexity might help determine if this inefficiency is unavoidable or can be improved.
- Concrete examples for some geometric problems:
  - MAXIMUM INDEPENDENT SET for unit disk graphs
  - MINIMUM VERTEX COVER for unit disk graphs
  - COVERING POINTS WITH SQUARES

Approximation schemes

Polynomial-Time Approximation Scheme (PTAS): algorithm that produces an ϵ-approximate solution in time $n^{f(\epsilon)}$.

Example: $n^{O(1/\epsilon)}$ time PTAS for INDEPENDENT SET in unit disk graphs.

Approximation schemes

Polynomial-Time Approximation Scheme (PTAS): algorithm that produces an ϵ-approximate solution in time $n^{f(\epsilon)}$.
Example: $n^{O(1/\epsilon)}$ time PTAS for INDEPENDENT SET in unit disk graphs.

Efficient Polynomial-Time Approximation Scheme (EPTAS): algorithm that produces an ϵ-approximate solution in time $f(\epsilon) \cdot n^c$.
Example: $2^{O(1/\epsilon)} \cdot n$ time PTAS for INDEPENDENT SET in planar graphs.

Approximation schemes

Polynomial-Time Approximation Scheme (PTAS): algorithm that produces an ϵ-approximate solution in time $n^{f(\epsilon)}$.
Example: $n^{O(1/\epsilon)}$ time PTAS for INDEPENDENT SET in unit disk graphs.

Efficient Polynomial-Time Approximation Scheme (EPTAS): algorithm that produces an ϵ-approximate solution in time $f(\epsilon) \cdot n^c$.
Example: $2^{O(1/\epsilon)} \cdot n$ time PTAS for INDEPENDENT SET in planar graphs.

Fully Polynomial-Time Approximation Scheme (FPTAS): algorithm that produces an ϵ-approximate solution in time $(1/\epsilon)^c \cdot n^c$.
Example: $O(1/\epsilon \cdot n^3)$ time PTAS for KNAPSACK.

The PTAS ~~scandal~~ situation

Running time of some approximation schemes for 20% error:
(reproduced from [Downey '03])

MULTIPLE KNAPSACK [Checkuri and Khanna '00]	$O(n^{9,375,000})$
MAXIMUM SUBFOREST [Shamir and Tsur '98]	$O(n^{958,267,391})$
GENERAL MULTIPROCESSOR JOB SCHEDULING [Chen and Miranda '99]	$> O(n^{10^{60}})$ (4 processors)
MAXIMUM INDEPENDENT SET for disk graphs [Erlebach *et al.* '01]	$O(n^{523,804})$

PTAS vs. EPTAS

Do people care whether their PTAS is an EPTAS?

- Arora's first algorithm for Euclidean TSP [FOCS '96] has running time $n^{O(1/\epsilon)}$ $\Rightarrow$ it is not an EPTAS.
  The algorithm in the journal version [*J. ACM* '98] is improved to $n \cdot \log^{O(1/\epsilon)} n = 2^{O(1/\epsilon^2)} \cdot n^2$ time $\Rightarrow$ it is an EPTAS.

PTAS vs. EPTAS

Do people care whether their PTAS is an EPTAS?

- Arora's first algorithm for Euclidean TSP [FOCS '96] has running time $n^{O(1/\epsilon)}$ $\Rightarrow$ it is not an EPTAS.
  The algorithm in the journal version [*J. ACM* '98] is improved to $n \cdot \log^{O(1/\epsilon)} n = 2^{O(1/\epsilon^2)} \cdot n^2$ time $\Rightarrow$ it is an EPTAS.
- Hunt *et al.* [*J. Alg.* '98] presented an $n^{O(1/\epsilon)}$ time PTAS for INDEPEDENT SET in unit disk graphs, and an $2^{O(1/(\epsilon\lambda)^2)} \cdot n$ time EPTAS for the special case of λ-precision unit disk graphs.

PTAS vs. EPTAS

Do people care whether their PTAS is an EPTAS?

- Arora's first algorithm for Euclidean TSP [FOCS '96] has running time $n^{O(1/\epsilon)}$ $\Rightarrow$ it is not an EPTAS.
  The algorithm in the journal version [*J. ACM* '98] is improved to $n \cdot \log^{O(1/\epsilon)} n = 2^{O(1/\epsilon^2)} \cdot n^2$ time $\Rightarrow$ it is an EPTAS.
- Hunt *et al.* [*J. Alg.* '98] presented an $n^{O(1/\epsilon)}$ time PTAS for INDEPEDENT SET in unit disk graphs, and an $2^{O(1/(\epsilon\lambda)^2)} \cdot n$ time EPTAS for the special case of λ-precision unit disk graphs.
- Arora *et al.* [STOC '98] gave an $n^{O(1/\epsilon)}$ time PTAS for the Euclidean k-median problem.
  Kolliopoulos and Rao [ESA '99] gave a $2^{O(1/\epsilon \cdot \log 1/\epsilon)} \cdot n \log^6 n$ time EPTAS for the problem.

PTAS vs. EPTAS

Do people care whether their PTAS is an EPTAS?

- Arora's first algorithm for Euclidean TSP [FOCS '96] has running time $n^{O(1/\epsilon)}$ $\Rightarrow$ it is not an EPTAS.
  The algorithm in the journal version [*J. ACM* '98] is improved to $n \cdot \log^{O(1/\epsilon)} n = 2^{O(1/\epsilon^2)} \cdot n^2$ time $\Rightarrow$ it is an EPTAS.
- Hunt *et al.* [*J. Alg.* '98] presented an $n^{O(1/\epsilon)}$ time PTAS for INDEPEDENT SET in unit disk graphs, and an $2^{O(1/(\epsilon\lambda)^2)} \cdot n$ time EPTAS for the special case of λ-precision unit disk graphs.
- Arora *et al.* [STOC '98] gave an $n^{O(1/\epsilon)}$ time PTAS for the Euclidean k-median problem.
  Kolliopoulos and Rao [ESA '99] gave a $2^{O(1/\epsilon \cdot \log 1/\epsilon)} \cdot n \log^6 n$ time EPTAS for the problem.

? When is such an improvement possible? Can we prove negative results?

Parameterized complexity

Goal: restrict the exponential growth of the running time to one parameter of the input.

Example: Finding a vertex cover of size k can be done in $O(1.3^k \cdot n^2)$ time.
Example: Finding a path of length k can be done in $O(2^k \cdot n^2)$ time.
Example: No algorithm with running time $n^{o(k)}$ is known for finding a k-clique.

Parameterized complexity

Goal: restrict the exponential growth of the running time to one parameter of the input.

Example: Finding a vertex cover of size k can be done in $O(1.3^k \cdot n^2)$ time.
Example: Finding a path of length k can be done in $O(2^k \cdot n^2)$ time.
Example: No algorithm with running time $n^{o(k)}$ is known for finding a k-clique.

In a **parameterized problem**, every instance has a special part k called the **parameter.**

Definition: A parameterized problem is **fixed-parameter tractable (FPT)** with parameter k if there is an algorithm with running time $f(k) \cdot n^c$ where c is a fixed constant not depending on k.

Parameterized intractability

We expect that MAXIMUM INDEPENDENT SET is not fixed-parameter tractable, no $n^{o(k)}$ algorithm is known.

W[1]-complete $\approx$ “as hard as MAXIMUM INDEPENDENT SET”

Parameterized intractability

We expect that MAXIMUM INDEPENDENT SET is not fixed-parameter tractable, no $n^{o(k)}$ algorithm is known.

W[1]-complete $\approx$ “as hard as MAXIMUM INDEPENDENT SET”

Parameterized reductions: L_1 is reducible to L_2, if there is a function f that transforms (x, k) to (x', k') such that

- $(x, k) \in L_1$ if and only if $(x', k') \in L_2$,
- f can be computed in $f(k)|x|^c$ time,
- **k' depends only on k**

If L_1 is reducible to L_2, and L_2 is in FPT, then L_1 is in FPT as well.

Most NP-completeness proofs are not good for parameterized reductions.

Relation to approximability

Optimization problem $\Rightarrow$ Parameterized decision problem:
"Is there a solution with value k?"

Observation: [Bazgan '95; Cesati and Trevisan '97] If there is an EPTAS for the optimization problem, then the decision problem is FPT.

Relation to approximability

Optimization problem $\Rightarrow$ Parameterized decision problem:
"Is there a solution with value k?"

Observation: [Bazgan '95; Cesati and Trevisan '97] If there is an EPTAS for the optimization problem, then the decision problem is FPT.

Proof: Given an $f(\epsilon) \cdot n^c$ time EPTAS, set $\epsilon = \frac{1}{2k}$, now the approximation algorithm can decide in $f(\frac{1}{2k}) \cdot n^c$ time whether the optimum is k or $k+1$.

Relation to approximability

Optimization problem $\Rightarrow$ Parameterized decision problem:
"Is there a solution with value k?"

Observation: [Bazgan '95; Cesati and Trevisan '97] If there is an EPTAS for the optimization problem, then the decision problem is FPT.

Proof: Given an $f(\epsilon) \cdot n^c$ time EPTAS, set $\epsilon = \frac{1}{2k}$, now the approximation algorithm can decide in $f(\frac{1}{2k}) \cdot n^c$ time whether the optimum is k or $k+1$.

Consequence: If the decision problem is W[1]-hard, then there is no EPTAS for the optimization problem, unless W[1]=FPT.

Relation to approximability

Optimization problem $\Rightarrow$ Parameterized decision problem:
"Is there a solution with value k?"

Observation: [Bazgan '95; Cesati and Trevisan '97] If there is an EPTAS for the optimization problem, then the decision problem is FPT.

Proof: Given an $f(\epsilon) \cdot n^c$ time EPTAS, set $\epsilon = \frac{1}{2k}$, now the approximation algorithm can decide in $f(\frac{1}{2k}) \cdot n^c$ time whether the optimum is k or $k+1$.

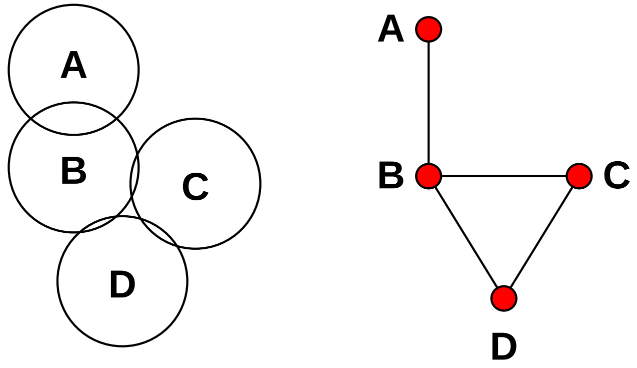
Consequence: If the decision problem is W[1]-hard, then there is no EPTAS for the optimization problem, unless W[1]=FPT.

Remark: Does not work the other way. Problem is FPT does not imply that there is an EPTAS. For example, MINIMUM VERTEX COVER can be solved in $1.3^k \cdot n^2$ time, but there is no PTAS.

Geometric problems

Geometric problems: problems involving geometric objects (usually in 2D or 3D). Often motivated by practical applications.

Geometric graphs: Intersection graphs of geometric objects. Vertices are the objects, two vertices are connected if the objects intersect.
Examples: disk graphs, unit disk graphs, coin graphs (=planar graphs).



Classical problems such as INDEPENDENT SET, DOMINATING SET, VERTEX COVER, VERTEX COLORING were investigated on different types of geometric graphs.

INDEPENDENT SET *for unit disk graphs*

Unit disk graphs:

- INDEPENDENT SET is NP-hard for unit disk graphs [Clark *et al.* '90].
- Admits an $n^{O(1/\epsilon)}$ time PTAS [Hunt *et al.* '98] and an $n^{O(1/\epsilon^2 \cdot \log 1/\epsilon)}$ time PTAS [Nieberg *et al.* '04].
- Parameterized problem can be solved in $n^{\sqrt{k}}$ time [Alber and Fiala '03].

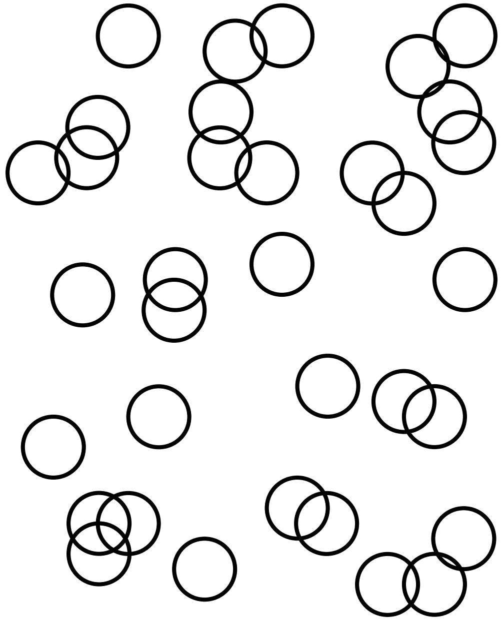
INDEPENDENT SET *for unit disk graphs*

Unit disk graphs:

- INDEPENDENT SET is NP-hard for unit disk graphs [Clark *et al.* '90].
- Admits an $n^{O(1/\epsilon)}$ time PTAS [Hunt *et al.* '98] and an $n^{O(1/\epsilon^2 \cdot \log 1/\epsilon)}$ time PTAS [Nieberg *et al.* '04].
- Parameterized problem can be solved in $n^{\sqrt{k}}$ time [Alber and Fiala '03].
- **New result:** parameterized version is W[1]-hard
  $\Rightarrow$ no EPTAS (unless W[1]=FPT).

PTAS for INDEPENDENT SET

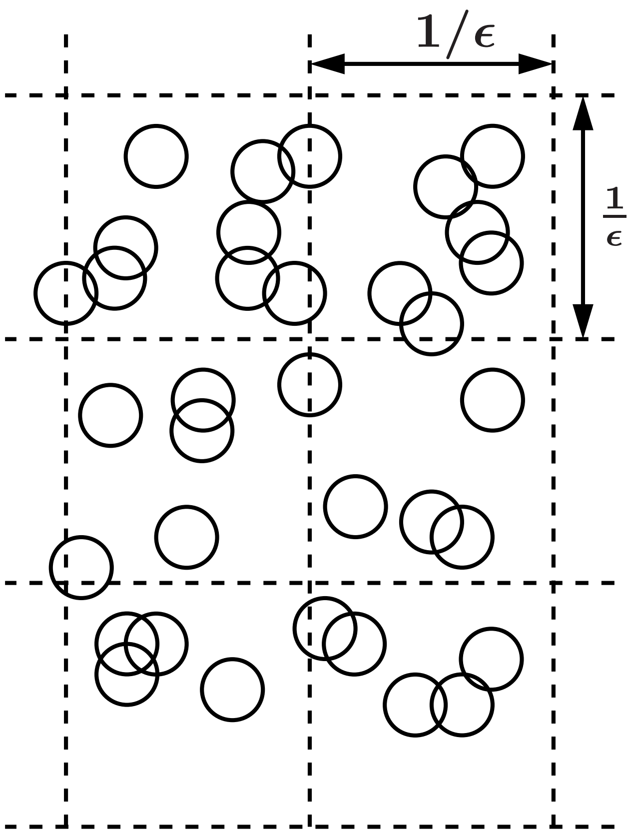
Sketch of the algorithm [Hunt *et al.* '98]:



PTAS for INDEPENDENT SET

Sketch of the algorithm [Hunt *et al.* '98]:

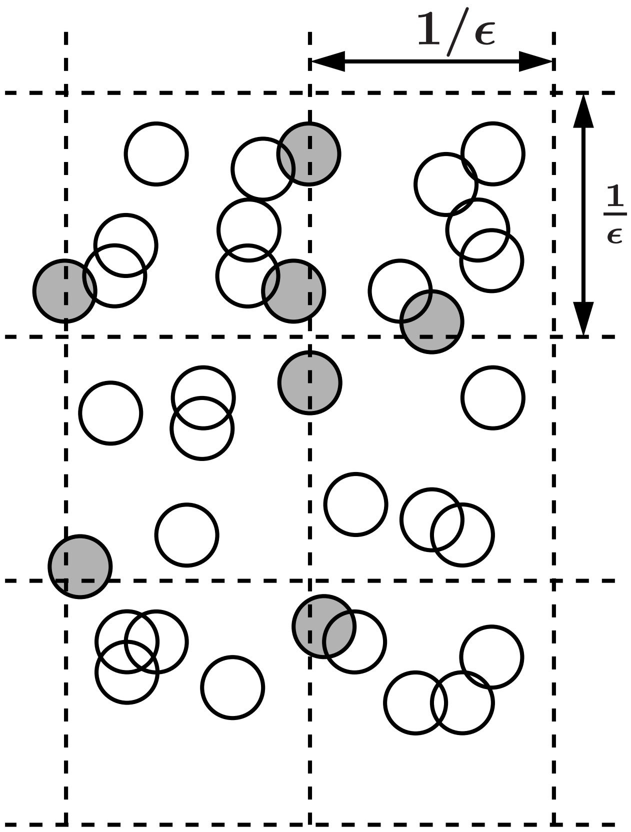
- Draw parallel horizontal and vertical lines.



PTAS for INDEPENDENT SET

Sketch of the algorithm [Hunt *et al.* '98]:

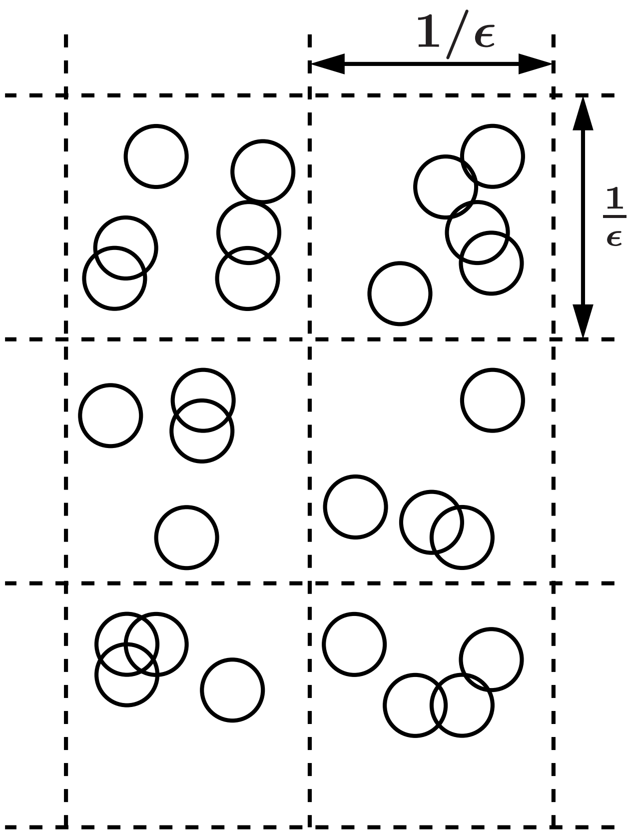
- Draw parallel horizontal and vertical lines.



PTAS for INDEPENDENT SET

Sketch of the algorithm [Hunt *et al.* '98]:

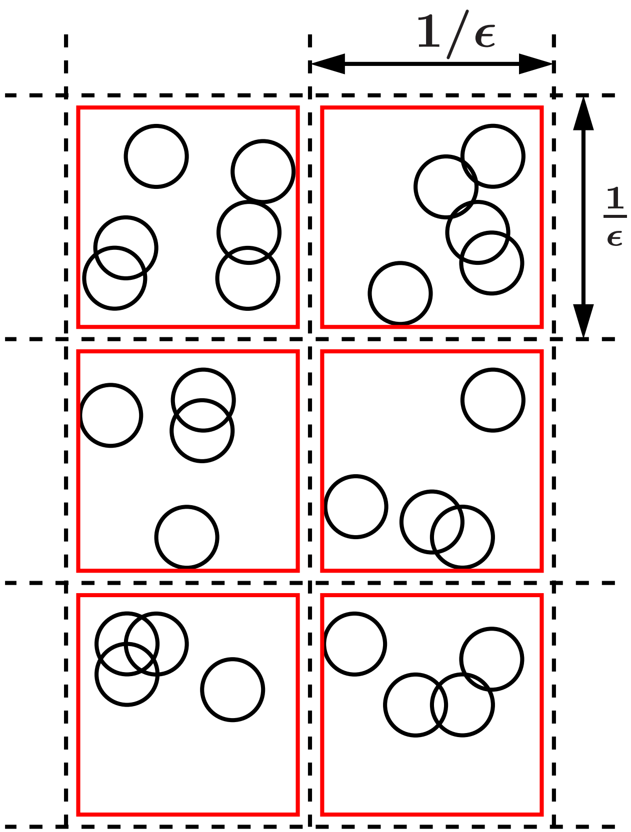
- Draw parallel horizontal and vertical lines.
- Remove the disks hit by the lines.



PTAS for INDEPENDENT SET

Sketch of the algorithm [Hunt *et al.* '98]:

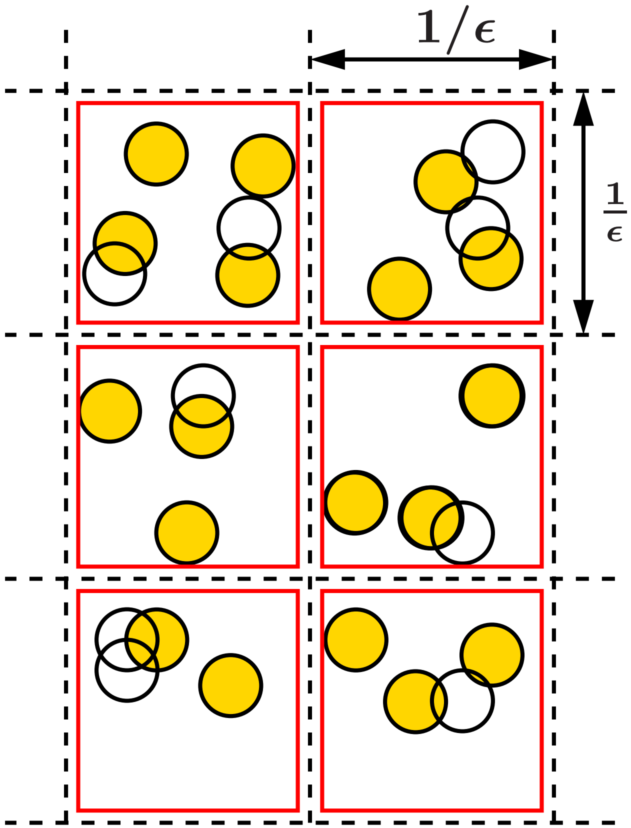
- Draw parallel horizontal and vertical lines.
- Remove the disks hit by the lines.
- Remaining problem breaks into small independent parts. . .



PTAS for INDEPENDENT SET

Sketch of the algorithm [Hunt *et al.* '98]:

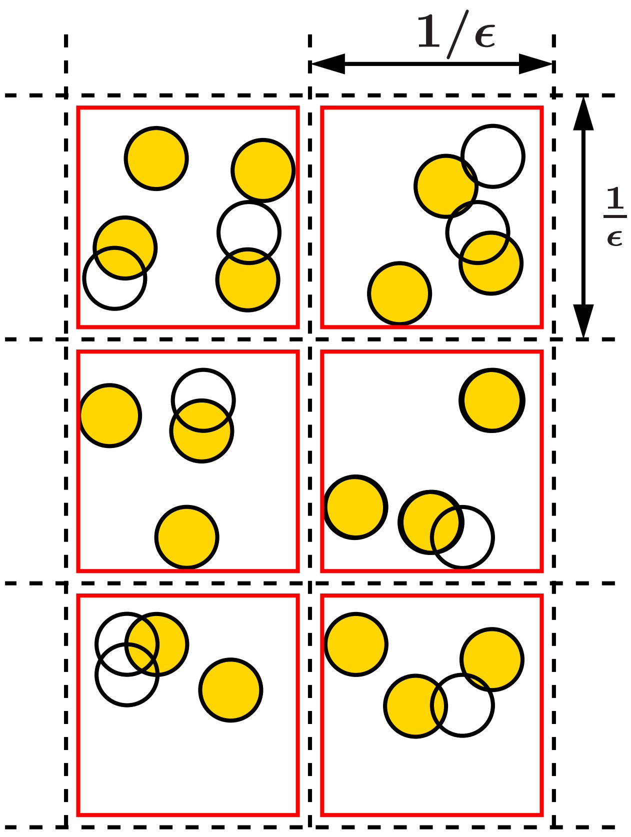
- Draw parallel horizontal and vertical lines.
- Remove the disks hit by the lines.
- Remaining problem breaks into small independent parts...
- ...that can be solved by brute force in $n^{O(1/\epsilon^2)}$ time.



PTAS for INDEPENDENT SET

Sketch of the algorithm [Hunt *et al.* '98]:

- Draw parallel horizontal and vertical lines.
- Remove the disks hit by the lines.
- Remaining problem breaks into small independent parts...
- ...that can be solved by brute force in $n^{O(1/\epsilon^2)}$ time.
- Shifting argument: there is at least one way of drawing the lines such that deleting the disks does not change the optimum much.



λ-precision unit disk graphs

λ-precision unit disk graphs (distance of centers is at least λ):

- INDEPENDENT SET is NP-hard even for $\lambda = 1$.
- Admits an EPTAS [Hunt *et al.* '98]. (Each small problem contains at most $O(1/(\lambda\epsilon)^2)$ disks.)
- Can be solved in $2^{\sqrt{k}} + n^c$ time [Alber and Fiala '03] $\Rightarrow$ FPT.

VERTEX COVER *for unit disk graphs*

[Hunt *et al.* '98] modifies the PTAS for INDEPENDENT SET to obtain an $n^{O(1/\epsilon)}$ time PTAS for VERTEX COVER.

Is there an EPTAS for the problem? VERTEX COVER is FPT, hence we cannot prove negative results.

VERTEX COVER *for unit disk graphs*

[Hunt *et al.* '98] modifies the PTAS for INDEPENDENT SET to obtain an $n^{O(1/\epsilon)}$ time PTAS for VERTEX COVER.

Is there an EPTAS for the problem? VERTEX COVER is FPT, hence we cannot prove negative results.

There is an EPTAS:

- Split the problem into $1/\epsilon \times 1/\epsilon$ rectangles.
- If a point is covered by more than $1/\epsilon$ disks, then select all these disks into the vertex cover (all but one has to be selected anyway).
- If every point is covered by at most $1/\epsilon$ disks $\Rightarrow$ there are at most $O(1/\epsilon^3)$ disks $\Rightarrow$ can be solved by brute force in $f(\epsilon)$ time.

COVERING POINTS WITH SQUARES

Given: n points in the plane

Find: a minimum number of unit squares such that every point is covered by at least one square.

$n^{O(1/\epsilon^2)}$ time PTAS by Hochbaum and Maas [*J. ACM* '85]. (Usual shifting strategy: split the problem into small parts and use brute force.)

Parameterized version: "Is it possible to cover the points with k squares?"

New result: the parameterized version of the problem is W[1]-hard

$\Rightarrow$ there is no EPTAS (unless FPT=W[1])

Conclusions

- PTASs are polynomial for fixed ϵ, but often with very high degree.
- PTAS vs. EPTAS
- Parameterized complexity can give evidence that there is no EPTAS for the problem.
- Works especially well for geometric problems.
- Concrete examples: INDEPENDENT SET and VERTEX COVER for unit disk graphs, COVERING POINTS WITH SQUARES.