Polynomial-time approximation schemes for NP-hard geometric problems

Reto Spöhel

max planck institut informatik

or

On Euclidean vehicle routing with allocation

Jan Remy, Reto Spöhel, Andreas Weißl

(appeared in WADS ’07, CGTA ’11)

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The Traveling Salesman Problem

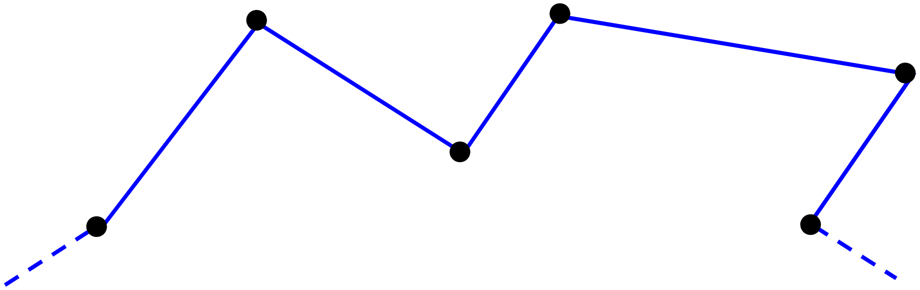
- **The Traveling Salesman Problem (TSP)**
  - Input: edge-weighted graph G
  - Output: Hamilton cycle in G with minimum edge-weight
- **Motivation:**
  - Traveling salesman ;-)
- **Complexity:**
  - NP-hard
  - Admits no constant factor approximation (unless P=NP) [Sahni and Gonzalez 76]

Metric TSP

- **Metric TSP**
  - Input: edge-weighted graph G satisfying triangle inequality
  - Output: Hamilton cycle in G with minimum edge-weight
- **Motivation:**
  - real-world problems usually satisfy triangle inequality
- **Complexity:**
  - still NP-hard
  - admits 3/2-approximation [Christofides 76]
  - admits no PTAS (unless P=NP) [Arora *et al.* 98]

Euclidean TSP

- **Euclidean TSP**
  - Input: points P $\subset \mathbb{R}^2$
  - Output: tour π through P with minimal length
- **Complexity:**
  - still NP-hard [Papadimitriou 77]
  - admits PTAS [Arora 96; Mitchell 96]



Euclidean TSP

- **Euclidean TSP**
  - Input: points $P \subset \mathbb{R}^2$
  - Output: tour π through P with minimal length
- **Complexity:**
  - still NP-hard [Papadimitriou 77]
  - admits PTAS [Arora 96; Mitchell 96]

Arora (FOCS '97)

There is a randomized PTAS for Euclidean TSP with complexity $n \log^{O(1/\epsilon)} n$.

  - …even one with complexity O(n log n).

Rao, Smith (STOC '98)

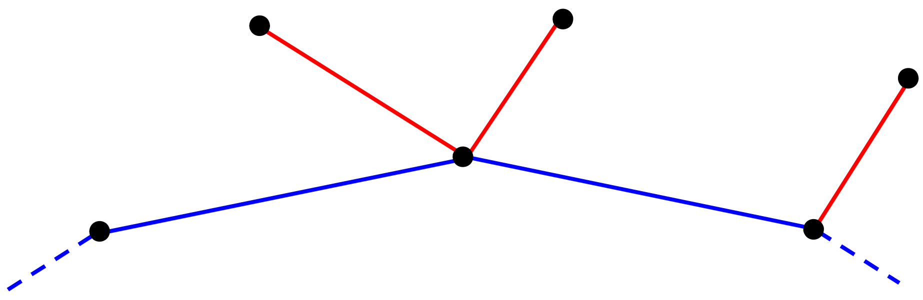
There is a randomized PTAS for Euclidean TSP with complexity O(n log n).

VRAP

- **(Euclidean) Vehicle Routing with Allocation (VRAP)**
  - Input: points P $\subset \mathbb{R}^2$, constant $\beta \geq$ 1
  - Output: tour π through subset T $\subseteq$ P minimizing

$$\underbrace{\sum_{\{p,q\}\in\pi} d(p,q)}_{\text{tour length}} + \beta \cdot \underbrace{\sum_{p\in P\setminus T} \min_{q\in T} d(p,q)}_{\text{allocation cost}}$$

- **Motivation:**
  - salesman does not visit all customers
  - customers not visited go to next tourpoint, which is more expensive by a factor of β.



VRAP

- **(Euclidean) Vehicle Routing with Allocation (VRAP)**
  - Input: points P $\subset \mathbb{R}^2$, constant $\beta \geq 1$
  - Output: tour π through subset T $\subseteq$ P minimizing

$$\underbrace{\sum_{\{p,q\}\in\pi} d(p,q)}_{\text{tour length}} + \beta \cdot \underbrace{\sum_{p\in P\setminus T} \min_{q\in T} d(p,q)}_{\text{allocation cost}}$$

- **Complexity:**
  - NP-hard, since setting $\beta \geq 2$ yields Euclidean TSP
  - as for Euclidean TSP, there exists a quasilinear PTAS

Remy, S., Weißl (WADS '07)

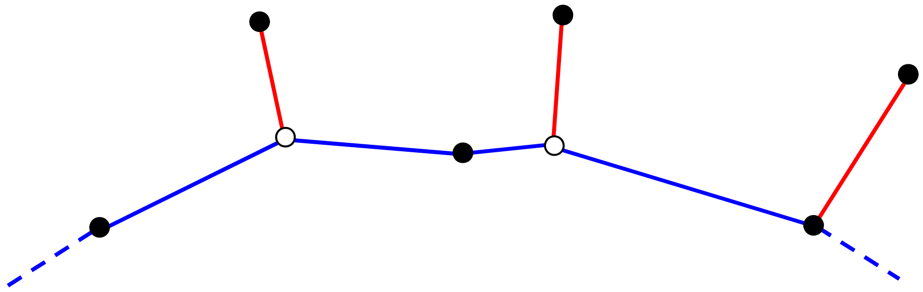
There is a randomized PTAS for VRAP with complexity $O(n \log^4 n)$.

Steiner VRAP

- **Steiner VRAP**
  - Input: points P $\subset \mathbb{R}^2$, constant $\beta \geq 1$
  - Output: subset T $\subseteq$ P, set of points S $\subset \mathbb{R}^2$ (Steiner Points), tour π through T $\cup$ S minimizing

$$\underbrace{\sum_{\{p,q\}\in\pi} d(p,q)}_{\text{tour length}} + \underbrace{\beta \cdot \sum_{p\in P\setminus T} \min_{q\in T\cup S} d(p,q)}_{\text{allocation cost}}$$

- **Motivation:**
  - salesman may also stop en route to serve customers



Steiner VRAP

- **Steiner VRAP**
  - Input: points $P \subset \mathbb{R}^2$, constant $\beta \geq 1$
  - Output: subset $T \subseteq P$, set of points $S \subset \mathbb{R}^2$ (Steiner Points), tour π through $T \cup S$ minimizing …
- **Complexity:**
  - NP-hard
  - admits PTAS

Armon, Avidor, Schwartz (ESA '06)

There is a randomized PTAS for Steiner VRAP with complexity $n^{O(1/\epsilon)}$.

  - …even a quasilinear one

Remy, S., Weißl (WADS '07)

There is a randomized PTAS for Steiner VRAP with complexity $n \log^{O(1/\epsilon)} n$.

Techniques

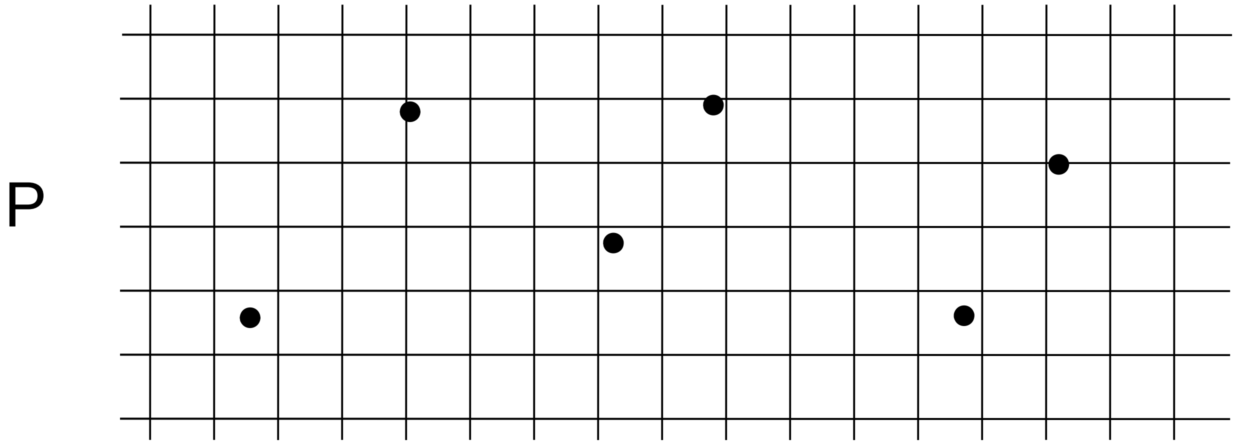
- Finding a good solution for VRAP means
  - a) finding a good set of tour points $T \subseteq P$
  - b) finding a good tour on this set T

  simultaneously.

- **a)** is essentially a facility location problem. 3
  - We use the adaptive dissection technique, due to [Kolliopoulos and Rao, ESA '99]
- **b)** is Euclidean TSP. 2
  - We use dynamic programming on 'patched short spanners', due to [Rao and Smith, STOC '98]
- To put both ideas into perspective, we start by explaining the basics of dynamic programming in quadtrees, as introduced in [Arora, FOCS '96] for Euclidean TSP 1

Preliminaries

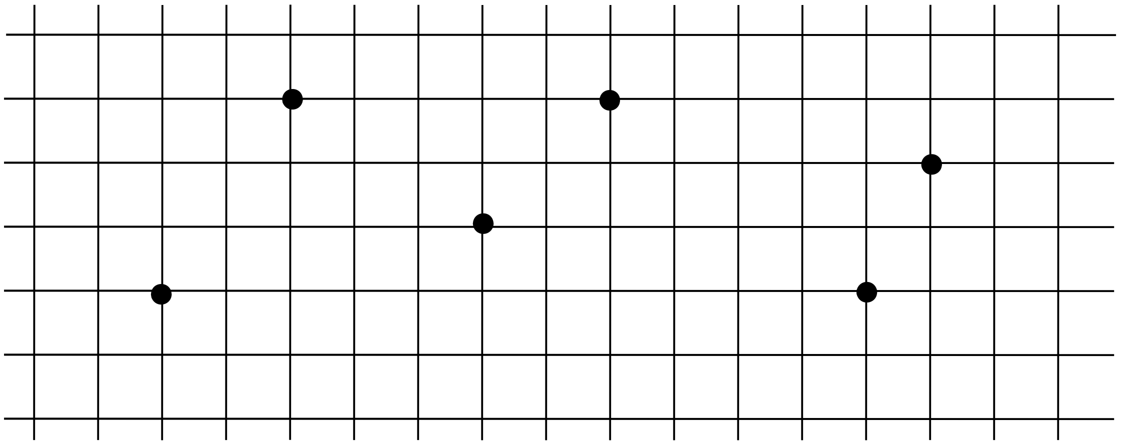
- We assume that the input points P
  - have odd integer coordinates
  - lie inside a square whose sidelength is
    - a power of 2
    - of order $O(n/\epsilon)$
- This is ok, since every $(1+\epsilon/2)$-approximation for the rescaled and shifted input P' corresponds to a $(1+\epsilon)$-approximation for the original input P.



Preliminaries

- We assume that the input points P
  - have odd integer coordinates
  - lie inside a square whose sidelength is
    - a power of 2
    - of order $O(n/\epsilon)$
- This is ok, since every $(1+\epsilon/2)$-approximation for the rescaled and shifted input P' corresponds to a $(1+\epsilon)$-approximation for the original input P.

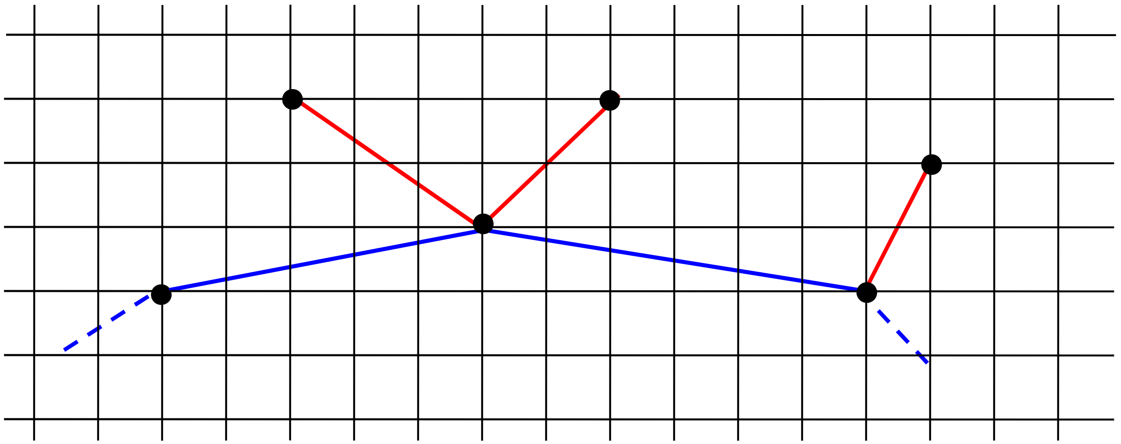
P'



Preliminaries

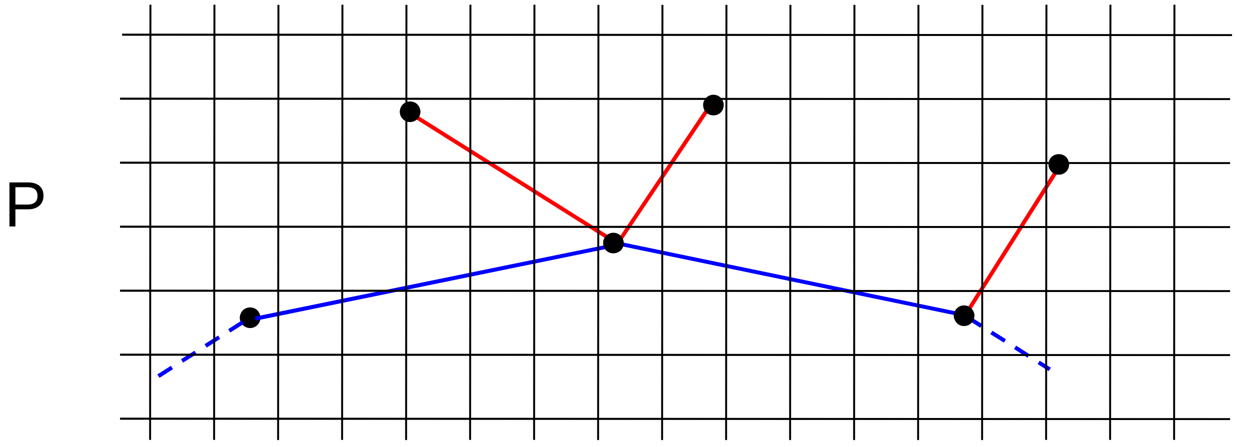
- We assume that the input points P
  - have odd integer coordinates
  - lie inside a square whose sidelength is
    - a power of 2
    - of order $O(n/\epsilon)$
- This is ok, since every $(1+\epsilon/2)$-approximation for the rescaled and shifted input P' corresponds to a $(1+\epsilon)$-approximation for the original input P.

P'



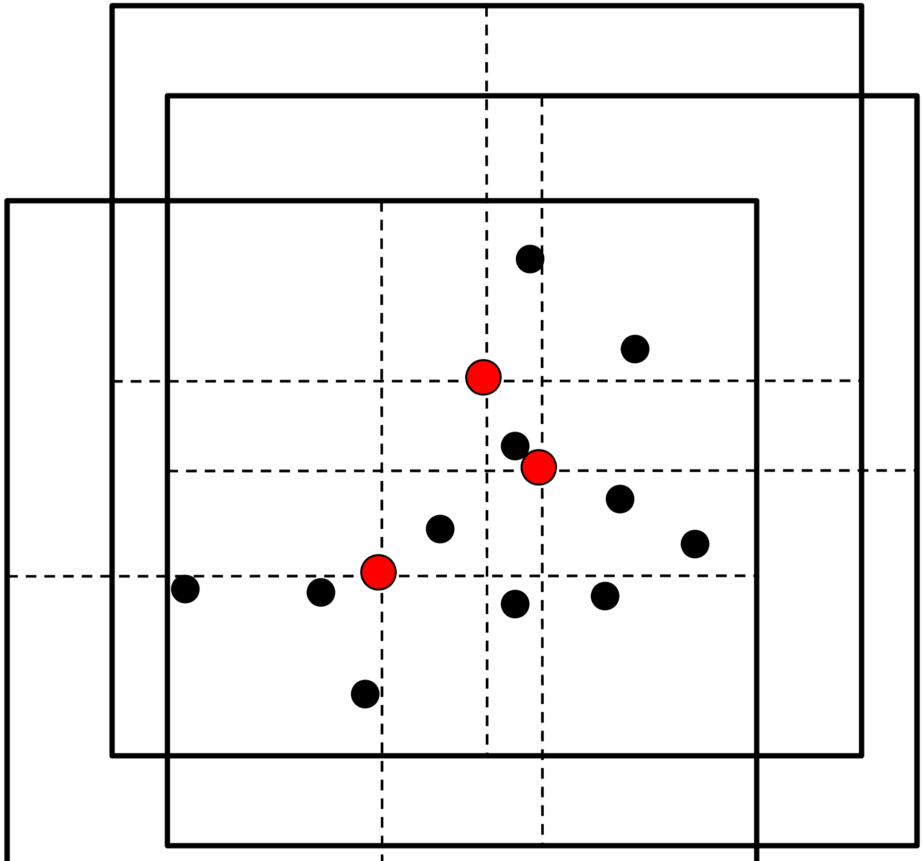
Preliminaries

- We assume that the input points P
  - have odd integer coordinates
  - lie inside a square whose sidelength is
    - a power of 2
    - of order $O(n/\epsilon)$
- This is ok, since every $(1+\epsilon/2)$-approximation for the rescaled and shifted input P' corresponds to a $(1+\epsilon)$-approximation for the original input P.



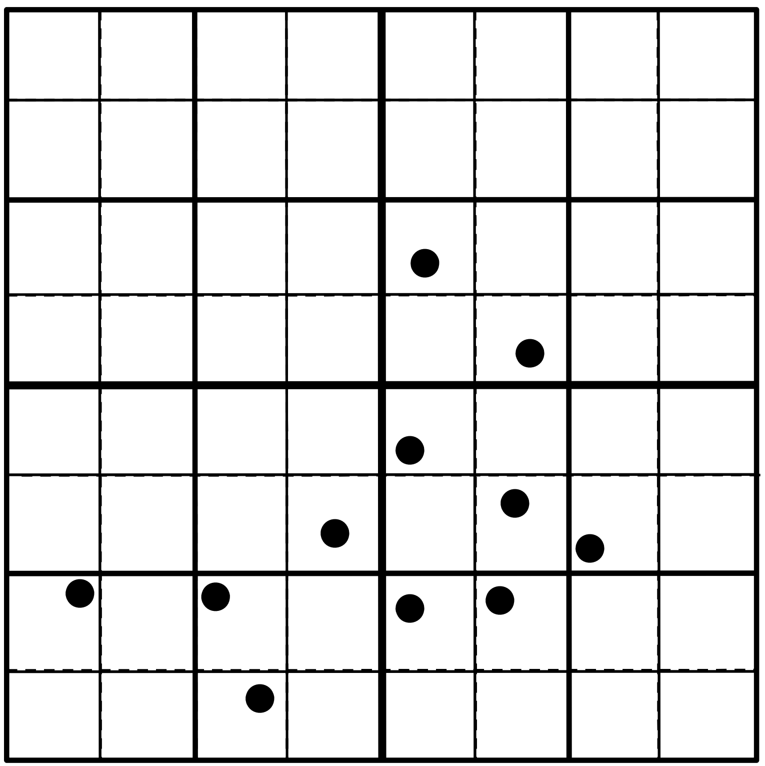
Quadtrees

- Choose origin of coordinate system (= center of large square) randomly.
  - this is the only source of randomness in all algorithms



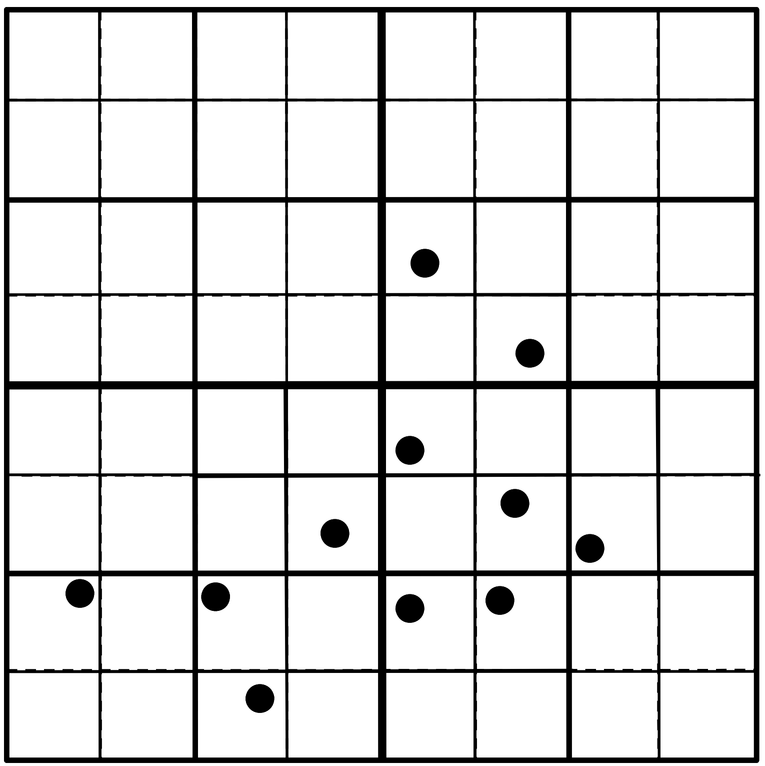
Quadtrees

- Split large square recursively into 4 smaller squares until squares have sidelength 2
  - Since bounding square has sidelength O(n), resulting tree has $O(n^2)$ nodes (squares) and depth O(log n)



Quadtrees

- Truncated quadtree: stop subdivision at empty squares
  - remaining tree has O(n log n) nodes

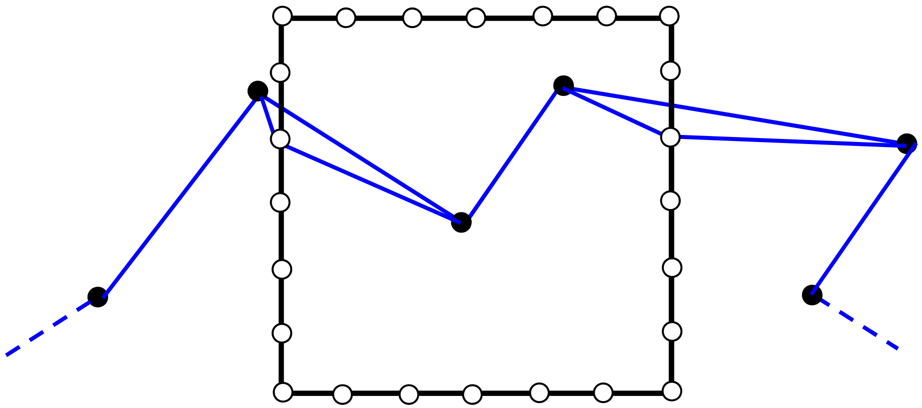


Portal-respecting solutions

- Place O(log n/ϵ) many equidistant points ('portals') on the boundary of each square.
  - **Impose restriction**: Salesman may enter/leave a square only via its portals.

Lemma (Arora)

In expectation, detouring all edges of the optimal salesman tour via the nearest portal increases its length only by a factor of 1+ϵ.



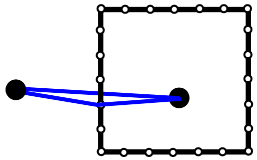
Portal-respecting solutions

- Place O(log n/ϵ) many equidistant points ('portals') on the boundary of each square.
  - **Impose restriction**: Salesman may enter/leave a square only via its portals.

Lemma (Arora)

In expectation, detouring all edges of the optimal salesman tour via the nearest portal increases its length only by a factor of 1+ϵ.

- **Intuition:** for two fixed points:
  - good



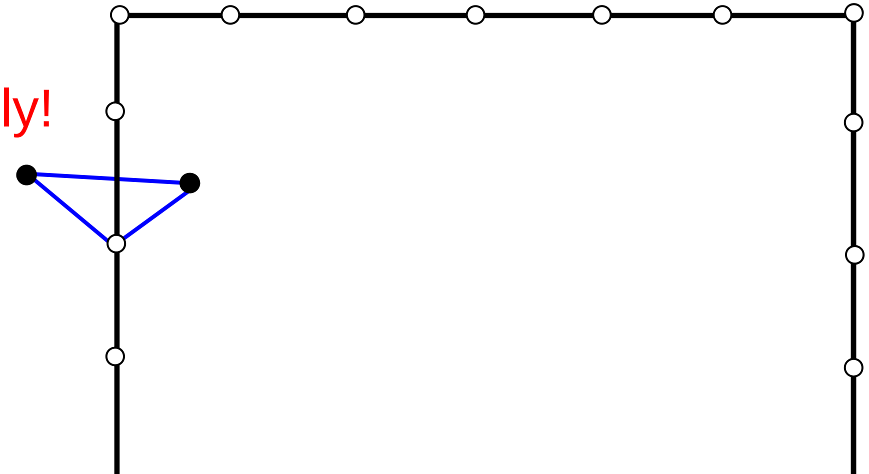
Portal-respecting solutions

- Place O(log n/ϵ) many equidistant points ('portals') on the boundary of each square.
  - **Impose restriction**: Salesman may enter/leave a square only via its portals.

Lemma (Arora)

In expectation, detouring all edges of the optimal salesman tour via the nearest portal increases its length only by a factor of $1+\epsilon$.

- **Intuition:** for two fixed points:
  - bad
  - but unlikely!



Portal-respecting solutions

- Place O(log n/ϵ) many equidistant points ('portals') on the boundary of each square.
  - **Impose restriction**: Salesman may enter/leave a square only via its portals.

Lemma (Arora)

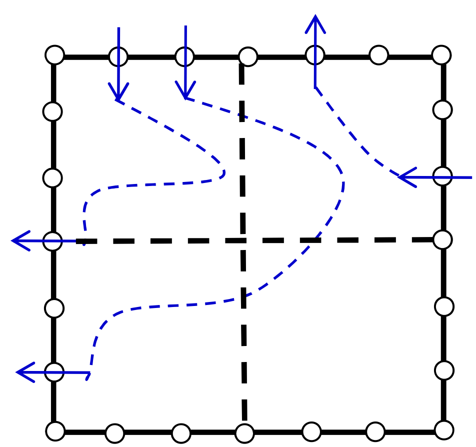
In expectation, detouring all edges of the optimal salesman tour via the nearest portal increases its length only by a factor of 1+ϵ.

  - i.e., there is an expected nearly-optimal portal-respecting salesman tour.
- We try to find the best portal-respecting salesman tour by dynamic programming in the quadtree.

Dynamic programming in quadtrees

- For a given square $\mathfrak{Q}$, guess which portals are used by salesman tour, and enumerate all possible configurations $\mathcal{C}$.
- For each configuration $\mathcal{C}$, calculate estimate for the length of a good tour inside $\mathfrak{Q}$, subject to the restrictions given by $\mathcal{C}$:
  - If $\mathfrak{Q}$ is a leaf of the quadtree, by brute force.
  - If $\mathfrak{Q}$ is an inner node of the quadtree, by recursing to its four children.

$\mathcal{C}$



Running time

- Working in a non-truncated quadtree, we have to consider $O(n^2)$ squares. For each of these we have to consider $2^{O(\log n/\epsilon)} = n^{O(1/\epsilon)}$ configurations, and the estimate for each configuration can be calculated in time $n^{O(1/\epsilon)}$.
  - We obtain a PTAS with running time

$$O(n^2) \cdot n^{O(1/\epsilon)} \cdot n^{O(1/\epsilon)} = \underline{\underline{n^{O(1/\epsilon)}}}$$

Arora (FOCS '96)

There is a randomized PTAS for Euclidean TSP with complexity $n^{O(1/\epsilon)}$.

- This is essentially the technique used in the PTAS for Steiner VRAP by Armon *et al*.

Armon, Avidor, Schwartz (ESA '06)

There is a randomized PTAS for Steiner VRAP with complexity $n^{O(1/\epsilon)}$.

Running time

- Working in a non-truncated quadtree, we have to consider $O(n^2)$ squares. For each of these we have to consider $2^{O(\log n/\epsilon)} = n^{O(1/\epsilon)}$ configurations, and the estimate for each configuration can be calculated in time $n^{O(1/\epsilon)}$.
  - We obtain a PTAS with running time

$$O(n^2) \cdot n^{O(1/\epsilon)} \cdot n^{O(1/\epsilon)} = \underline{\underline{n^{O(1/\epsilon)}}}$$

Arora (FOCS '96)

There is a randomized PTAS for Euclidean TSP with complexity $n^{O(1/\epsilon)}$.

- to achieve quasilinear time, we can only use polylogarithmic time per square. In particular, we can only consider polylogarithmically many configurations per square.

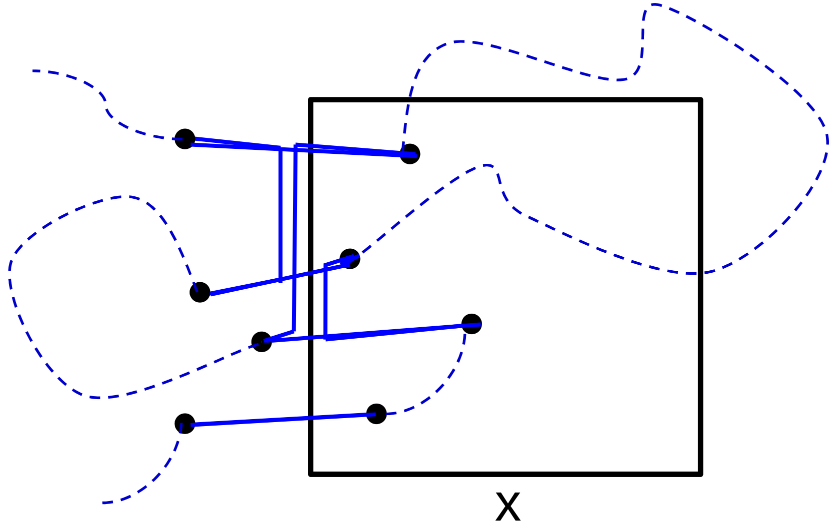
Improving the running time

Patching Lemma (Arora)

The optimal solution can be modified such that it crosses the boundary of every square at most $O(1/\epsilon)$ many times.

In expectation, this increases the length of the tour only by a factor of $1+\epsilon$.

- **Idea:** proceed bottom-up through quadtree and modify each square with too many crossings by introducing line segments parallel to sides.



- The total length of the new line segments is at most 3x
- ➔ modification on low levels of the quadtree are cheap.

Improving the running time

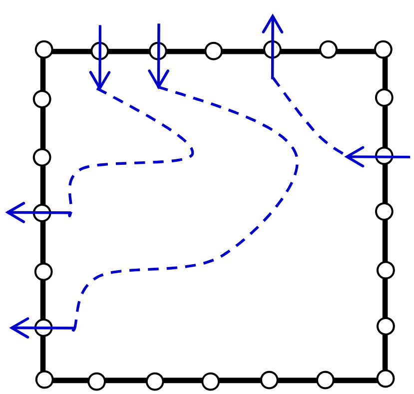
Patching Lemma (Arora)

The optimal solution can be modified such that it crosses the boundary of every square at most $O(1/\epsilon)$ many times.

In expectation, this increases the length of the tour only by a factor of $1+\epsilon$.

- i.e., there is an expected nearly-optimal portal-respecting salesman tour which for every square uses only $O(1/\epsilon)$ many of the $O(\log n)$ portals.

- Looking for such a 'patched' solution, we only have to consider $O(\log n)^{O(1/\epsilon)} = \log^{O(1/\epsilon)} n$ configurations per square!



Improving the running time

Patching Lemma (Arora)

The optimal solution can be modified such that it crosses the boundary of every square at most $O(1/\epsilon)$ many times.
In expectation, this increases the length of the tour only by a factor of $1+\epsilon$.

- We only have to consider $\log^{O(1/\epsilon)} n$ configurations per square.
  - Working in a truncated quadtree, we obtain a PTAS with running time

    $$O(n \log n) \cdot \log^{O(1/\epsilon)} n \cdot \log^{O(1/\epsilon)} n = \underline{\underline{n \log^{O(1/\epsilon)} n}}$$

Arora (FOCS '97)

There is a randomized PTAS for Euclidean TSP with complexity $n \log^{O(1/\epsilon)} n$.

Improving the running time

Patching Lemma (Arora)

The optimal solution can be modified such that it crosses the boundary of every square at most $O(1/\epsilon)$ many times.

In expectation, this increases the length of the tour only by a factor of $1+\epsilon$.

Lemma

The Patching Lemma extends to Steiner VRAP.

- Combining the extended patching lemma with standard quadtree techniques for facility location problems [Arora, Raghavan, Rao, STOC '98], we obtain

Remy, S., Weißl (WADS '07)

There is a randomized PTAS for Steiner VRAP with complexity $n \log^{O(1/\epsilon)} n$.

Improving the running time even further

2

- **Patching revisited:**
  - In Arora's technique, the 'patching' is not part of the algorithm – we simply know a nearly-optimal patched solution exists, and try to find it by dynamic programming.
  - Rao and Smith (STOC '98) improved Arora's running time by making the 'patching' part of the algorithm.
- Effect: We only have to consider constantly many configurations per square!
  - Yields a PTAS with running time

$$O(n \log n) \cdot O(1) \cdot O(1) = \underline{\underline{O(n \log n)}}$$

Rao, Smith (STOC '98)

There is a randomized PTAS for Euclidean TSP with complexity O(n log n).

Improving the running time even further

Remy, S., Weißl (WADS '07)

There is a randomized PTAS for (non-Steiner) VRAP with complexity $O(n \log^4 n)$.

- Combine the $O(n \log n)$ technique for Euclidean TSP with a clever technique for the facility location part.
- […]

2 3

- Concluding remarks:
  - All algorithms can be derandomized trivially at the cost of an extra factor $O(n^2)$.
  - All algorithms generalize to higher dimensions (with increased, but still polynomial running times).

Summary

- VRAP is a combination of Euclidean TSP and a facility location problem.

- The two state-of-the-art techniques
  - Dynamic programming on ‘patched short spanners’ (Rao and Smith, STOC ’98) for Euclidean TSP
  - Adaptive dissection (Kolliopoulos and Rao, ESA ’99) for facility location

  can be combined into a $O(n \log^4 n)$-PTAS for VRAP.

Thank you!

Questions?