Approximation Algorithms

Ruben Hoeksma (University of Bremen)

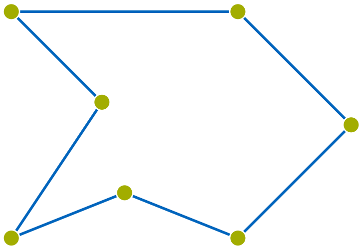
June 18, 2019

Polynomial time approximation scheme for Euclidean TSP

Euclidean TSP

Input: n points in the Euclidean space ($\mathbb{R}^2$)

Goal: Find the shortest tour that visits all points (order the points).



Theorem (Arora '98)

There is a PTAS for TSP in the Euclidean plane.

Proof idea

Show that there is a close to optimal solution that satisfies nice properties, by transforming an optimal solution into one. Then, show that an optimal solution with such properties can be found in polynomial time.

Reminder: PTAS

For any $\varepsilon > 0$ gives $(1+\varepsilon)$-approximation. $1/\varepsilon$ is considered **constant** for the computation of the running time.

Main steps:

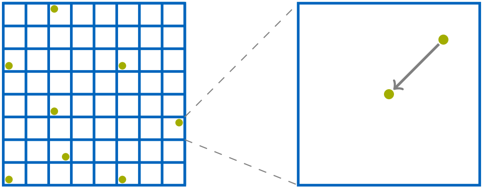
1. Simplify the problem.
2. Prove that simplification is not too bad.
3. Find optimal simplified solution.

- 1a. Round the instance.
- 1b. Restrict the set of solutions.
- 2. Randomization.
- 3. Dynamic programming.

1a. Rounding (perturbation)

Let B be the smallest axes-parallel box that contains all points. Scale the instance such that B is a square with size $(\frac{n}{\varepsilon})^2 \times (\frac{n}{\varepsilon})^2$. W.l.o.g., we assume that $\frac{n}{\varepsilon} = 2^{k'}$. Let $L = (\frac{n}{\varepsilon})^2 = 2^k$, for $k = 2\log_2 \frac{n}{\varepsilon}$.

Introduce a unit grid on B. We move each point of the input to the center of the unit grid square in which it is contained.



Lemma: An optimal order on the rounded instance is a $(1+\varepsilon)$-approximation in the original instance.

Proof

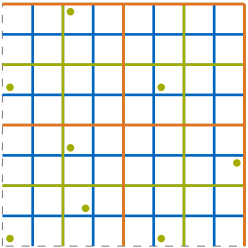
Exercise.

1b1. Quad tree dissection

Divide B into four boxes repeatedly.

Def. (Level i squares/lines)

Add $2 \times 2^{i-1}$ level i lines in step i for $i \geq 2$ (2 level 1 lines).
Squares created by level i lines are level i squares.



This divides B into different-level squares. B is the level 0 square.

- 4 level 1 squares, size $\frac{L}{2} \times \frac{L}{2}$.
- 16 level 2 squares, size $\frac{L}{4} \times \frac{L}{4}$.
- $\vdots$
- 4^i level i squares, size $\frac{L}{2^i} \times \frac{L}{2^i}$.
- L^2 level k squares, size 1×1.

Idea: squares are the cells of the dynamic programming table.

1b2. Portals

Let m= $\lceil \frac{k}{\varepsilon} \rceil$. For each i, place m equidistant portals on all sides of every level i square that is part of a level i line.

Def. (Portal respecting)

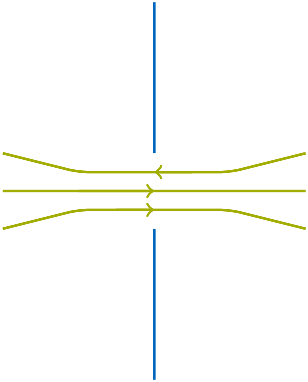
A tour is portal respecting if it crosses line only through portals.

Lemma

There is an optimal portal respecting tour crossing each portal at most twice.

Proof.

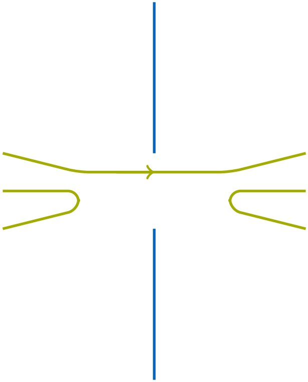
Suppose not, then there is a portal p that is crossed at least trice. Give the tour an orientation. p is crossed in the same direction at least twice. Cut those crosses and close them on both sides of the portal. This decreases the number of crosses by two.
Correctness $\rightarrow$ exercise. □



Proof.

Suppose not, then there is a portal p that is crossed at least trice. Give the tour an orientation. p is crossed in the same direction at least twice. Cut those crosses and close them on both sides of the portal. This decreases the number of crosses by two.

Correctness $\rightarrow$ exercise. $\square$

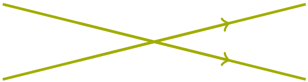


Lemma

An optimal tour does not self-intersect.

Proof.

If an optimal tour self-intersects, direct the tour, cut around the intersection and reconnect same direction to each other.
Correctness $\rightarrow$ exercise. $\square$



Lemma

An optimal tour does not self-intersect.

Proof.

If an optimal tour self-intersects, direct the tour, cut around the intersection and reconnect same direction to each other.
Correctness → exercise. □

Would be nice if ...

The optimal portal respecting tour is a $(1+\varepsilon)$-approximation.

Not true!!! The cost for going through the portals can be very high.

Problem: Extra distance to portals depends on the level of the line.

- A level 1 line has $2 \times m$ portals.
- A level 2 line has $4 \times m$ portals.
- A level i line has $2^i \times m$ portals.

Solution: 2. Shift the lines (randomly).

Move vertical line by integer a and horizontal lines by integer b. Thus a vertical line at x is now at $x + a \mod L$ and a horizontal line at y is now at $y + b \mod L$.

Shift the lines

Move vertical line by integer a and horizontal lines by integer b. Thus a vertical line at x is now at $x + a \mod L$ and a horizontal line at y is now at $y + b \mod L$.

Note: There are L^2 possible shifts.

Note 2: Some squares now wrap around B.

Lemma

Let T be an optimal tour. There is a shift (a, b) such that there is a portal respecting tour with length at most $(1 + 2\varepsilon)\text{Len}(T)$.

Proof

We prove this for a and b chosen uniformly at random and in expectation.
T consists of n straight lines. Therefore, T crosses at most $2 \cdot \mathrm{Len}(T)$ grid lines.
Let (a, b) be chosen uniformly at random from $\{1, \ldots, L\}^2$. The number and locations of the grid line crossings of T is independent of (a, b). Only the level of the grid lines crossed depends on (a, b).
For each grid line crossing independently, the probability of that being a level i line is at most

$$\frac{\#(\text{level } i \text{ lines})}{\#(\text{lines total})} = \frac{2 \cdot 2^i}{2L} = \frac{2^i}{L} \, .$$

The distance to a level i portal from any level i grid line crossing is at most

$$\frac{1}{2} \frac{L}{2^i m} \, .$$

Proof

In expectation the cost of a detour through a portal costs

$$\sum_{i=1}^{k} \frac{L}{2^i m} \frac{2^i}{L} = \sum_{i=1}^{k} \frac{1}{m} = \frac{k}{m} \leq \varepsilon \, .$$

Thus, the total cost of the tour T', that is T with portal detours, is in expectation at most

$$\mathrm{Len}(T) + 2\mathrm{Len}(T)\varepsilon = (1 + 2\varepsilon)\mathrm{Len}(T) \, .$$

□

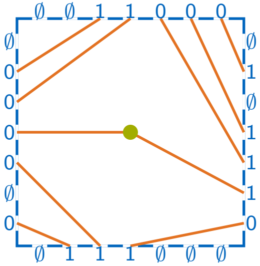
Conclusion: There is an (a, b) such that the bound holds.

3. Dynamic program

One dynamic programming cell for each combination of

- shift parameters (a, b);
- level i;
- level i square shifted by (a, b);
- pairs of connected portals.

Level k, 1×1 square:



$\leq 4m$ portals.
No self-intersections.
At most 2 passes through each portal.
Portal pairs are connected with straight lines.
One pair visits the input point.

Claim: There are 3^{8m} possible combinations of portal pairs and $(4m)^2$ choices how to connect the input point.

3. Dynamic program

Number of cells:

- $L^2 \in O(n^4/\varepsilon^4)$ parameters (a, b).
- $O(4^k) = O(n^2/\varepsilon^2)$ squares:
  - k levels.
  - 4^i squares at level i.
- $n^{O(1/\varepsilon)}$ combinations of portals.

Total number of cells is $n^{O(1/\varepsilon)}$.

Compute value of a cell:

At level k compute all options to visit the input point $((4m)^2)$.

At level i compute all combinations of the four level $i+1$ squares that are contained in the square that connect the right portals. There are $n^{O(1/\varepsilon)}$ options per level $i+1$ square, for a total of $\left(n^{O(1/\varepsilon)}\right)^4 = n^{O(1/\varepsilon)}$.

Conclusion: There are $n^{O(1/\varepsilon)}$ cells and computation of each costs $n^{O(1/\varepsilon)}$. Thus total running time is $n^{O(1/\varepsilon)}$. Finally, compare all level 0 square cells ($n^{O(1/\varepsilon)}$ many) and take the best one.