Arora's PTAS for Euclidean TSP

Andreas Galanis

April 30, 2009

Table of contents

1. Basics
   - Bounding box
   - Dissect!
   - Portals & well behaved tours
2. DP
   - Valid Visits
   - The algorithm
3. Bypassing losses
   - Counterexample
   - Randomize!
   - Losses Revisited

Introduction

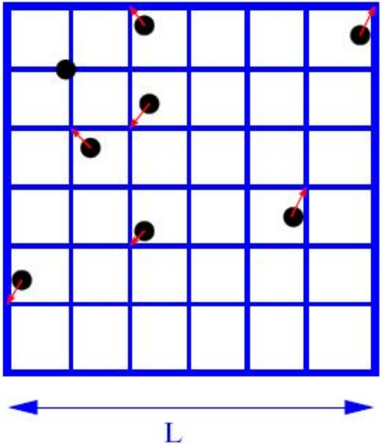
- Traveling Salesman Problem NP-complete
  - Hard to approximate.
- Metric TSP
  - Edge costs satisfy triangle inequality.
  - Factor 2 approximation algorithm in $O(m + n \log n)$.
  - Factor 3/2 approximation algorithm in $O(n^3)$.
- Euclidean TSP
  - Special case of Metric TSP.
  - Euclidean distance as cost function.
- Objective: Present a PTAS for Euclidean TSP.

Instance I

1. Consider n points in $\mathbb{R}^d$.
2. The graph is complete.
3. Euclidean distance $\mathrm{dist}(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d}(x_i - y_i)^2\right)^{1/2}$.
4. We consider the case $d = 2$, i.e. n points in the plane.
   - Most of it applies to the general case with slight modifications.

Transform I to I'

- Consider the smallest square that encloses all n points.
    - At least two nodes are on opposite edges of the square.
    - $OPT \geq 2L$.
- Set the length of each edge of the square to $L = 4n^2$.
    - Just a scale factor, so optimal tour is invariant
- Consider n to be a power of 2, so L is a power of 2 also, i.e. $L = 2^k$.
    - $k = 2 + 2\log n = O(\log n)$.
- Relocate every node of G to the nearest gridpoint.

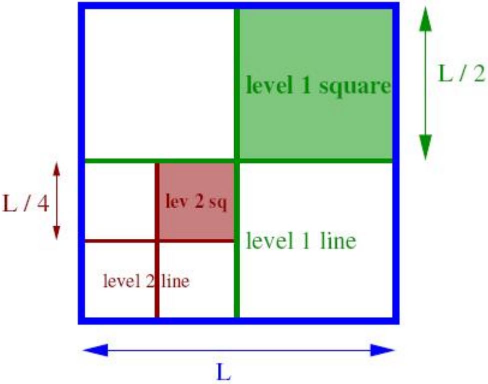


Solving I' is enough

- Maximum distance from arbitrary point to the nearest grid point is $\sqrt{2}/2$.
- Absolute error per node is $\sqrt{2}$, i.e. total absolute error $n\sqrt{2}$.
- $\dfrac{|SOL - OPT|}{OPT} \leq \dfrac{n\sqrt{2}}{2L} = \dfrac{n\sqrt{2}}{8n^2} = \dfrac{1}{4\sqrt{2}n}$.
- Thus, given a $(1+\epsilon)$-solution to I', the corresponding solution to I is $(1+\epsilon+\frac{1}{4\sqrt{2}n})$-approximate.
  - For sufficiently large n we can adjust ϵ to compensate for the relative error.

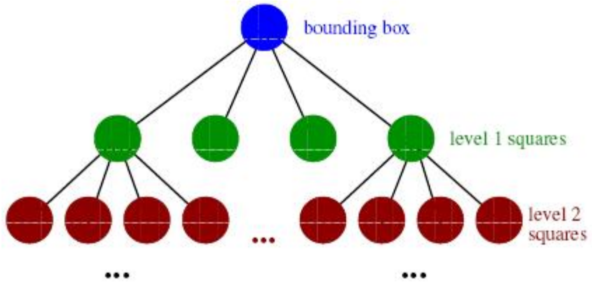
Basic dissection

- Partition the square with two lines into 4-subsquares.
- Recursively partition the resulting squares until unit squares are obtained.
- A level i square has size $L/2^i \times L/2^i$.



Basic dissection

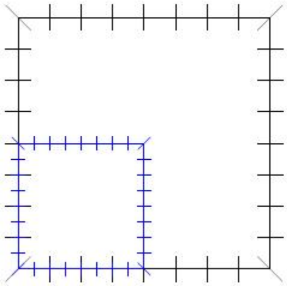
- Basic dissection can be seen as a 4-ary tree with depth k.



- Number of nodes
$1 + 4 + \ldots + 4^k = O(4^{k+1}) = O(4^{2+2\log n}) = O(n^4)$.

Portals

- Restrict the tour to intersect the level lines at certain points (*portals*).
- Each square has one portal for each corner and $m-1$ portals for each edge, i.e. all in all $4m$ portals for each square.

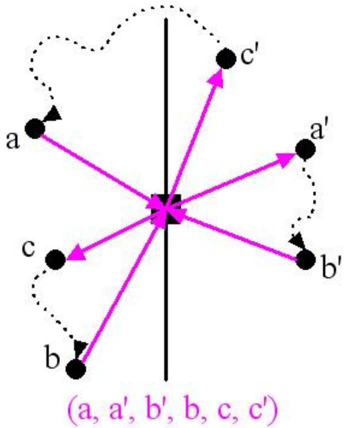


- Choose m a power of 2 in the interval $\left[\frac{k}{\epsilon}, \frac{2k}{\epsilon}\right]$.
- Level i portals are also level $i+1$ portals.

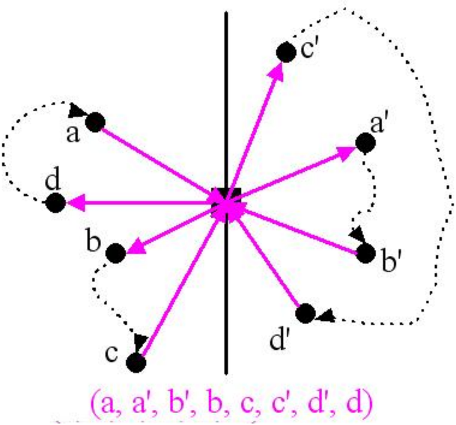
Well-behaved tours

1. A tour is *well-behaved* if it is a tour on the n points and any subset of the portals.
2. A tour is *well-behaved with limited crossings* if it is a well-behaved tour and visits each portal at most twice.
3. Claim: Any well-behaved tour can be transformed to a well-behaved tour with limited crossings without increasing its length.
4. Thus, it suffices (??) to search for well-behaved with limited crossings tours.
    - No guarantee though that a well-behaved tour is actully close enough to the optimum. In fact, there are counterexamples that prove the contrary.
    - This difficulty will be treated later.

Making crossings ≤ 2 : Odd number of crossings

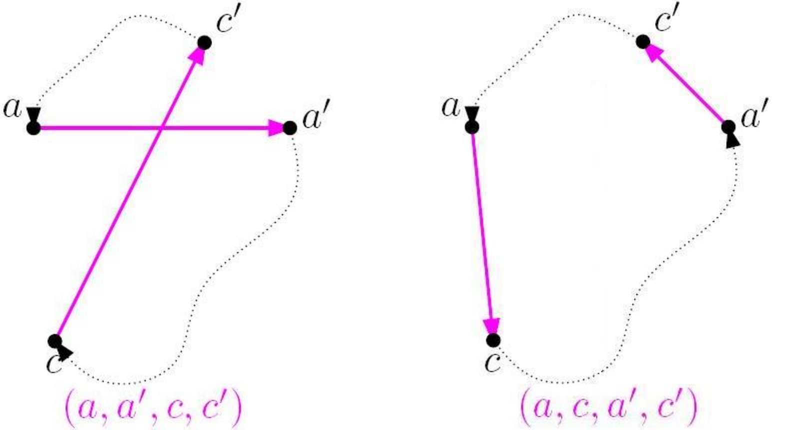


Making crossings ≤ 2 : Even number of crossings



Eliminating an intersection

We further restrict the tour to not intersect itself apart possibly a portal.



Triangle inequality guarantees that we haven't increased the length of the tour.

New objectives

We need to:

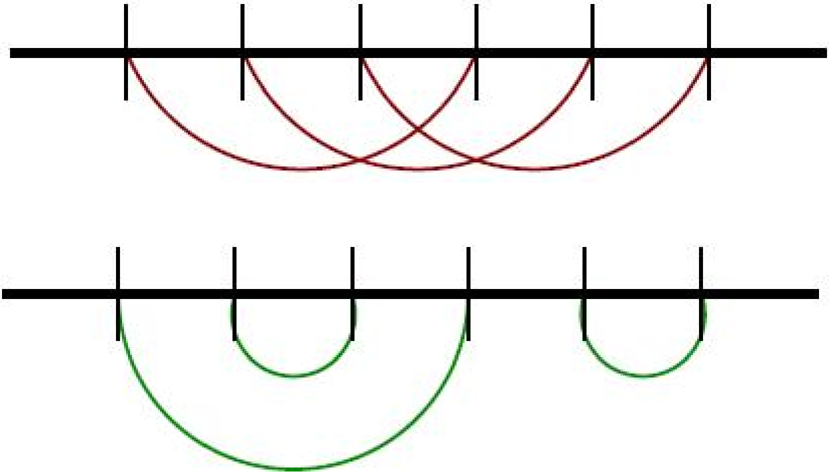
1. Find an optimal well-behaved tour with limited crossings.
2. Prove that this optimal tour is short enough.

We will use dynamic programming to fulfill the first goal.

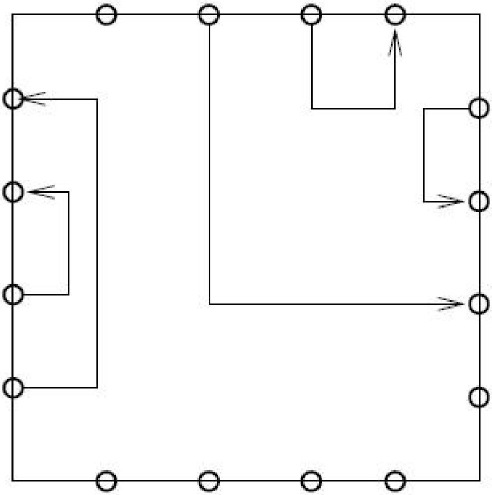
Dynamic Programming

- We look only for tours with limited crossings, i.e. each portal can be used 0, 1 or 2 times.
- $4m$ portals in total for each square, thus $3^{4m} = 2^{4m \log 3} = 2^{4k \log 3/\epsilon} = n^{O(1/\epsilon)}$ possibilities for each square.
- Once we have selected the portals, not every possible pairing is allowed because no self-intersection is allowed.

Valid pairings

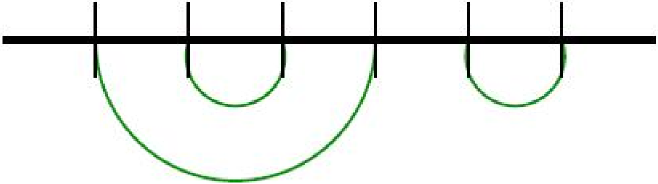


Valid pairings



Counting valid pairings

- There is a bijection between valid pairings with $2r$ portals and balanced arrangement of $2r$ parentheses.
- (())()



- The latter is the r-th Catalan number, $C_r = \frac{1}{r+1}\binom{2r}{r} < 2^{2r}$.
- Each visit on the portals of a square uses at most $8m$ portals, thus the number of valid pairings is at most

$$2^{8m} = 2^{8k/\epsilon} = n^{O(1/\epsilon)}.$$

Putting it together

- $n^{O(1/\epsilon)}$ possibilities of portal usage.
- $n^{O(1/\epsilon)}$ valid pairings for each portal usage.
- Total number of valid visits : $n^{O(1/\epsilon)} \cdot n^{O(1/\epsilon)} = n^{O(1/\epsilon)}$.

Dynamic Programming

minimum costs	level 0	level 1				...
	square 1	square 1	square 2	square 3	square 4	...
valid visit 1						
valid visit 2						
valid visit 3						
⋮						

$$\#\text{columns} = \#\text{nodes in 4-ary tree} = O(n^4).$$

$$\#\text{rows} = \#\text{valid visits} = n^{O(1/\epsilon)}.$$

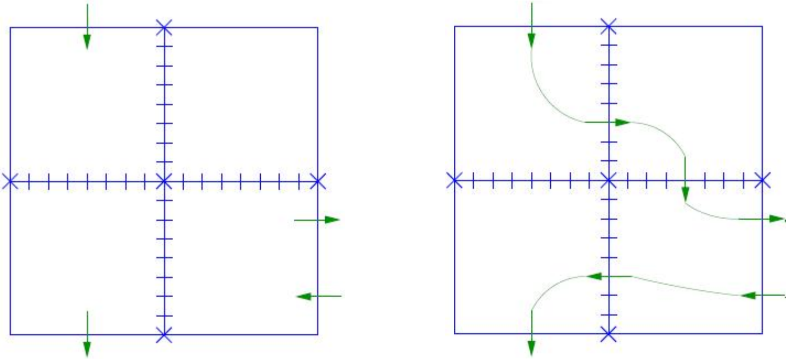
$$\#\text{entries} = n^{O(1/\epsilon)} \cdot n^{O(1/\epsilon)} = n^{O(1/\epsilon)}.$$

Dynamic Programming

minimum costs	level 0	level 1				. . .
	square 1	square 1	square 2	square 3	square 4	. . .
valid visit 1						
valid visit 2						
valid visit 3						
⋮						

1. Start at the leaves of the tree.
2. Use the results of the four children squares to compute the visits of the corresponding parent square.

Dynamic Programming - Cost per entry



- $4m+1$ internal portals, thus $3^{4m+1} = n^{O(1/\epsilon)}$ possible portal usage.
- Using again Catalan numbers, we obtain at most $2^{8m+2} = n^{O(1/\epsilon)}$ valid pairings.
- In total, we have $n^{O(1/\epsilon)}$ configurations to consider.

Dynamic Programming - Cost per entry

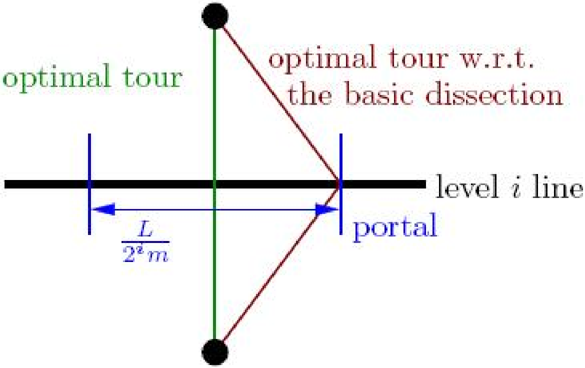
minimum costs	level 0	level 1				...
	square 1	square 1	square 2	square 3	square 4	...
valid visit 1						
valid visit 2						
valid visit 3						
⋮						

1. Sum the corresponding lengths of the appropriate visits of the children squares and find the minimum.
2. Total cost $=$ #entries $\cdot n^{O(1/\epsilon)} = n^{O(1/\epsilon)} \cdot n^{O(1/\epsilon)} = n^{O(1/\epsilon)}$.

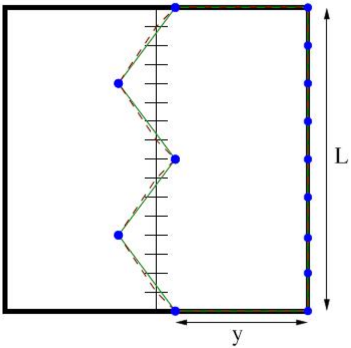
Losses

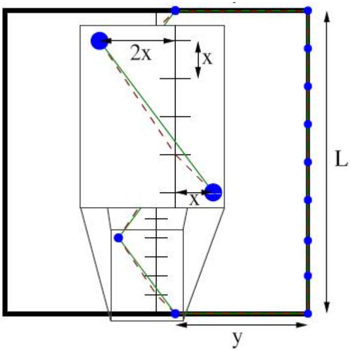
- We have computed the optimal well-behaved tour. Is it short enough?
- NO!

Why not?


optimal tour
optimal tour w.r.t. the basic dissection
level i line
portal
$\frac{L}{2^i m}$

Counterexample

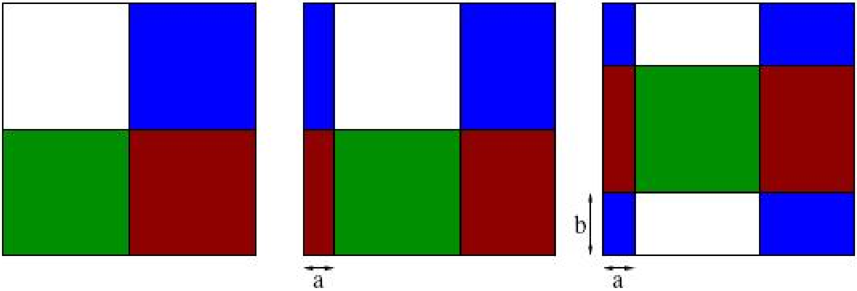




1. $OPT = \frac{\sqrt{3^2+4^2}}{4}L + 2y + L.$
2. $SOL = \frac{\sqrt{2}+\sqrt{2^2+3^2}}{4}L + 2y + L.$
3. $\frac{OPT}{SOL} > 1.0015.$

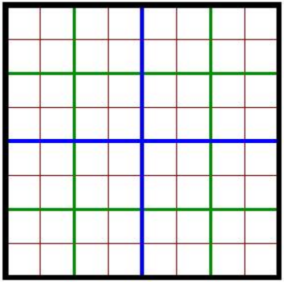
Shifted dissection

Choose randomly integers a, b such that $0 \leq a, b < L$ and shift each vertical line x to $x + a \mod L$ and each horizontal line y to $y + b \mod L$.



This way any specific line has random level.

Shifted dissection



1. 2^1 level 1 lines, 2^2 level 2 lines, ..., 2^k level k lines.
2. #lines$=2^{k+1}-1$.
3. Probability that a randomly chosen line has level i is

$$p(i) = \frac{2^i}{2^{k+1}-1} = \frac{2^i}{2L-2} \leq \frac{2^i}{L}.$$

Expected value of indirection

1. Maximum indirection when a level i line is crossed is

$$x(i) = \frac{L}{2^i m}$$

2. Expected value of indirection when a random line is crossed is

$$E(X) = \sum_{i=1}^{k} p(i)x(i) \leq \sum_{i=1}^{k} \frac{2^i}{L} \cdot \frac{L}{2^i m} = \frac{k}{m} \leq \epsilon$$

Expected value of total indirection

1. In order to find the expected value of total indirection we need to bound the number of crossings.
2. Let τ an optimal tour and let $N(\tau)$ the total number of crossings (both vertical and horizontal). Then

$$N(\tau) \leq \sqrt{2} \cdot OPT$$

3. If Y is the total indirection

$$E(Y) = N(\tau) \cdot E(X) \leq \sqrt{2} \cdot OPT \cdot \epsilon$$

4. Markov inequality implies

$$\Pr[Y \geq 2\sqrt{2}\epsilon \cdot OPT] \leq \frac{E(Y)}{2\sqrt{2}\epsilon \cdot OPT} \leq \frac{1}{2}$$

Error Bound

1. From the preceding, the probability that the error bound exceeds $2\sqrt{2}\epsilon$ is less than $1/2$. The $2\sqrt{2}$ constant can be tackled with a suitably chosen ϵ' $(2\sqrt{2}\epsilon' = \epsilon)$.
2. The algorithm can be derandomized by checking all the $O(L^2) = O(n^4)$ possibilities for a, b.

Proving $N(\tau) \leq \sqrt{2} \cdot OPT$

- Number of crossings equals the perimeter of the square (red line).
- $c^2 = a^2 + b^2$.
- $a^2 + b^2 \geq (a+b)^2/2$.
- $c\sqrt{2} \geq (a+b)$.
- Adding up we obtain the desired result.