PTAS for Euclidean TSP

K Vamsi Krishna

Computer Science and Automation,
Indian Institute of Science

6^{th} November, 2006

Outline

Approximating TSP

- **General TSP:** Cannot be approximated within any polynomial time computable factor $\alpha(n)$
- **Metric TSP:** Can be approximated within a constant factor, but has no PTAS
- **Euclidean TSP:** Can be approximated within any given factor

Problem

For fixed d, given n points in $\mathbf{R}^d$, the problem is to find the minimum length tour of the n points. The distance between any two points x and y is defined to be the Euclidean distance between them, i.e., $\left(\sum_{i=1}^{d}(x_i - y_i)^2\right)^{1/2}$.

Results

- Euclidean TSP is NP-Complete [3]
- The first PTAS was given by Arora in 1996 and shortly followed by Mitchell
- Comparison of running times

Paper	$\mathbf{R}^2$	$\mathbf{R}^d$
Arora '96 [1]	$n^{O(1/\epsilon)}$	$n^{(O((1/\epsilon)\log n))^{d-2}}$
Arora '97 [2]	$n(\log n)^{O(1/\epsilon)}$	$n(\log n)^{(O(\sqrt{d}/\epsilon))^{d-1}}$

Basic Idea

- Generally, the basic idea of a PTAS is to define a **coarse solution** depending on the error parameter ϵ and to find it using dynamic programming.
- The same idea appears in this algorithm. However, the coarse solution is specified **probabilistically**.

Overview

- **Main Idea:** Algorithm performs a recursive geometric partitioning of the instance and solves the subinstances thus produced using dynamic programming.
- **Algorithm: PTAS for Euclidean TSP**
  1. Make the input well-rounded by perturbation
  2. Build the quadtree with a random shift
  3. Compute the optimum portal-tour using DP
  4. Return the vertices in that order

Well-rounded Instance

Definition

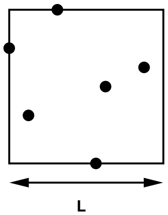
An instance is said to be **well-rounded** if it satifies the following properties.

1. All nodes have integral coordinates
2. Each (non-zero) internode distance is atleast 8 units
3. The maximum internode distance is $O(n)$

Bounding Box

Definition

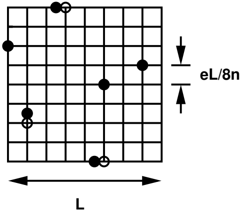
Bounding box is the smallest axis aligned square that contains all the input nodes.



- $c(T_{OPT}) = OPT$
- $OPT \geq L$

Perturbation Step (i): Snap to Grid

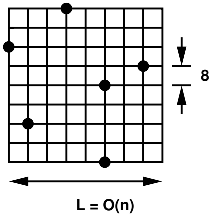
Consider a **grid** of granularity $\epsilon L/8n$ and move each node to it's nearest gridpoint.



- $\forall$ tours $T, |c(T) - c'(T)| \leq 2n\epsilon L/8n \leq \epsilon OPT/4$
- Should compute $(1+\epsilon')$-approximate tour

Perturbation Step (ii): Scale L to $O(n)$

Scale the distances by $s = \epsilon L/64n$.

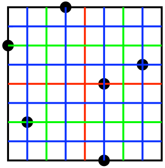


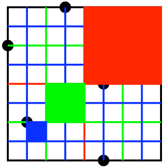
- $\forall$ tours $T, c'(T) = s.c(T)$
- There is no change in the approximation factor

Dissection

Definition

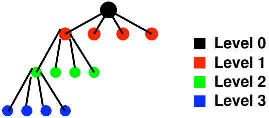
A **dissection** of the bounding box is a recursive partitioning into smaller squares, until they are of unit size.





Levels of Dissection

A dissection can be seen as a **4-ary tree**. Each square and the dissection line can be assigned a **level**.

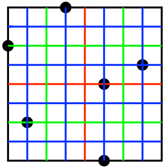


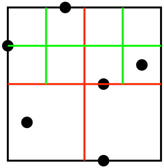
- Depth of the dissection $= O(\log n)$
- Number of leaves $= O(n^2)$
- Number of squares $= O(n^2)$
- Number of level i lines $= 2^i$
- Level i line forms the edge for squares at levels $i, i+1, \cdots$

Quadtree

Definition

A **quadtree** is defined similar to dissection, except that we stop the recursive partitioning as soon as the square has atmost one node.



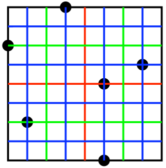


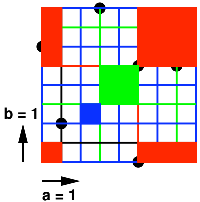
- Depth of the quadtree = $O(\log n)$
- Number of leaves with a node = n
- Number of squares = $O(n \log n)$

Shifted Dissection

Definition

Let a, b be integers. An (a, b)-**shifted dissection** is defined as the dissection obtained by **shifting** the dissection lines modulo L, where a is the shift for vertical lines and b is for horizontal lines.





- Some squares are **wrapped-around** in the shifted dissection
- Number of shifted dissections $= n^2$

Shifted Quadtree

Definition

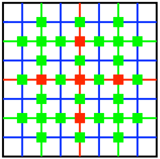
An (a, b)-**shifted quadtree** is obtained from the corresponding shifted dissection by cutting off the partitioning at squares that contain only 1 node.

- It can be constructed efficiently in $O(n \log^2 n)$ time

Portal

Definition

Portals are the designated points on the edges of the squares through which a tour can cross the boundary of any region in the dissection. Each square has a portal at each of its 4 corners and $m = O((1/\epsilon)\log L)$ equally spaced portals on each edge.

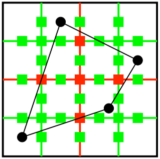


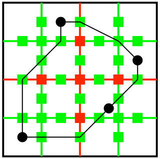
- Assume $L = 2^k$, for some k
- Choose $m = 2^{k'} - 1$, for some k'
- Portals at higher levels are also portals at lower levels

Portal-tour

Definition

A tour is said to be a **portal-tour** with respect to a dissection if it crosses each edge of each square in the dissection at most $r = O(c)$ times and always through one of the portals, however it is allowed to go through a portal multiple times.





- In terms of the original set of nodes, such a tour can be viewed as having **bent** edges. Note that straightening the bent edges at the end of the algorithm can only decrease the cost.

Structure Theorem

Theorem

For a randomly picked (a, b)-shifted dissection,

$$Pr[OPT_{(a,b)} \leq (1+\epsilon)OPT] \geq 1/2$$

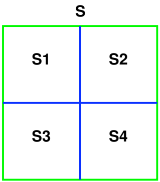
Dynamic Program

- DP can be use as **principle of optimality** holds
- Each subproblem is specified by
  - A square, $T = O(n \log n)$
  - Multiset of $\leq r$ portals for each edge, $\leq (m+3)^{4r}$
  - Pairing of the $\leq 4r$ portals specified above, $\leq (4r)!$
  - Portals and their pairing is called an **interface**
- Size of the lookup table $= O(T.(m+3)^{4r}.(4r)!)$

Dynamic Program (cont)

- Table is filled bottom-up starting at the leaves of the quadtree
- **Leaves** i.e., squares with atmost 1 node and $O(r)$ portals
  - Can be solved in $2^{O(r)}$ time
- **Internal nodes**
  - Enumerate interfaces for the 4 subsquares
  - Multiset of $\leq r$ portals for each edge, $\leq (m+3)^{4r}$
  - Traversal order of $\leq 4r$ portals specified above, $\leq (4r)^{4r}.(4r)!$
  - Number of interfaces enumerated $\leq (m+3)^{4r}.(4r)^{4r}.(4r)!$
  - Obtain cost corresponding to each subsquare and interface from the table
- Total running time $= O(T.(m+3)^{8r}.(4r)^{4r}.((4r)!)^2)$

Dynamic Program (cont)



- Interface I for S on the green edges is given
- Interfaces $I1, I2, I3, I4$ for $S1, S2, S3, S4$ on the blue egdes is enumerated

$$c(S, I) = MIN_{I1,I2,I3,I4}\big(c(S1, I1) + c(S2, I2) + c(S3, I3) + c(S4, I4)\big)$$

Overview

- Sufficient to prove the structure theorem
- Need for shifted dissection

Patching Lemma

Lemma

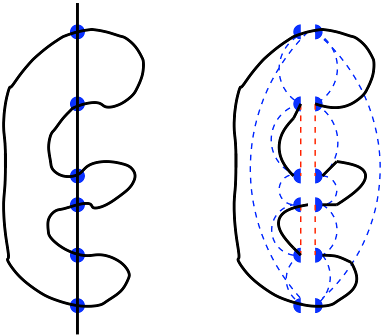
A tour π crossing a line S of length s more than thrice can be modified to cross at most twice, with an additional cost of $6s$.

Proof.

- Let number of times π crosses S be t
- Pick $2k$ crossing points on S
- Add segments of min cost tour through those points, $< 2s$
- Add segments of min cost matching among those points, $< s$
- Make two copies of all points and segments added above, $< 6s$
- Consider the Eulerian traversal of the points
- It crosses S at most twice

□

Patching Lemma (cont)



Crossing Lemma

Lemma

Let $t(\pi, l)$ be the number of times π crosses a dissection line l. Then $\sum_l t(\pi, l) \leq 2c(\pi)$.

Proof.

- Consider an edge of π of length s
- Let u and v be lengths of horizontal and vertical projections
- It constributes atmost $u + 1$ and $v + 1$ to the LHS
- $u^2 + v^2 = s^2$ and $u + v \leq \sqrt{2}s$
- $u + v + 2 \leq \sqrt{2}s + 2$
- $s \geq 8 \Rightarrow \sqrt{2}s + 2 \leq 2s$
- Summing over all the edges proves the lemma

$\square$

Structure Theorem

Theorem

For a randomly picked (a,b)-shifted dissection,

$$Pr[OPT_{(a,b)} \leq (1+\epsilon)OPT] \geq 1/2$$

Proof (Outline)

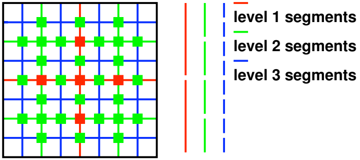
Let $g = 6$ be the constant from the patching lemma.

- Modify the optimum tour π into portal respecting wrt dissection
- $E_{a,b}$[increase in cost due to line l] $\leq 3g.t(\pi, l)/r$
- By linearity of expectation
  $E_{a,b}$[increase in cost] $\leq \sum_l 3g.t(\pi, l)/r$
  $\leq 6g.OPT/r$
  $\leq \epsilon.OPT/2$ if $r \geq 12g/\epsilon$
- By Markov inequality the theorem follows

Modification Step (i): Reduce Crossings

We shall prove the bound for vertical lines. Same arguments apply for horizontal lines.

- $Pr_a[l \text{ is at level } i] = 2^i/L$
- Size of level i square $= L/2^i$
- A level i line l forms an edge for squares at levels $j = i, i+1, \cdots,$ level j segment
- For each segment, the number of crossings should be atmost r



Modification Step (i): Reduce Crossings (cont)

Modify(l, i, b) For each segment (starting from smaller to larger) which has more than r crossings apply the patching lemma to reduce it to 4.

- Let $c(j)$ be the number of level j segments for which patching is applied
- $\sum_j c(j) \leq t(\pi, l)/(r-3)$, why?
- Increase in cost $\leq \sum_j c(j).g.\frac{L}{2^j}$ (by patching lemma)
- $Pr_a[l \text{ is at level } i] = 2^i/L$
- $E_a[\text{ increase in cost }]$

$$\leq \sum_{i \geq 1} \frac{2^i}{L} . \sum_{j \geq i} c(j).g.\frac{L}{2^j} = g.\sum_{j \geq 1} \frac{c(j)}{2^j} . \sum_{i \leq j} 2^i$$

$$\leq g.\sum_{j \geq 1} 2.c(j) \leq \frac{2g.t(\pi, l)}{r-3}$$

Modification Step (i): Reduce Crossings (cont)

- E_a[total increase in cost] $\leq \frac{2gt(\pi,l)}{r-3} + \frac{t(\pi,l)}{2r}$
  $\leq 3gt(\pi, l)/r$, when $r > 15$
- Patching on line l may increase the crossings on some horizontal line, however it is not a problem, why?

Modification Step (ii): Move to Portals

- Move each crossing on line l to the nearest portal
- Interportal distance on a level i line $= L/2^i m$
- Increase in cost per crossing $= L/2^i m$
- Number of crossings $= t(\pi, l)$
- $Pr_a[\ l$ is at level $i\ ] = 2^i/L$
- $E_a[$ increase in cost $] \leq \sum_i \frac{2^i}{L}.t(\pi, l).\frac{L}{2^i m}$
  $= t(\pi, l) \log L/m$
  $\leq t(\pi, l)/2r$, when $m \geq 2r \log L$

Conclusion

- Recap of the algorithm
- Higher dimensions
- Other norms
- Other geometric problems
- Thank you

References

Sanjeev Arora.
Polynomial time approximation schemes for euclidean TSP and other geometric problems.
In *FOCS: IEEE Symposium on Foundations of Computer Science*, 1996.

Sanjeev Arora.
Nearly linear time approximation schemes for euclidean TSP and other geometric problems.
In *FOCS: IEEE Symposium on Foundations of Computer Science*, 1997.

Christos H. Papadimitriou.
The euclidean traveling salesman problem is NP-complete.
Theoretical Computer Science, 4(3):237–244, 1977.

Effect of Moving to Grid Points

OPT tour (value) may change, however OPT' is not very far from OPT.

- $|OPT - OPT'| \leq \epsilon OPT/4$
- $OPT' \leq A' \leq (1 + x)OPT'$
- $|A - A'| \leq \epsilon OPT/4$
- $OPT \leq A \leq (1 + \epsilon)OPT$

$x = 2\epsilon/(4 + \epsilon)$, paper gives $x = 3\epsilon/4$.