CMPUT 675: Approximation Algorithms and Approximability **Winter 2018**

Lecture 14 (March 1): Euclidean TSP

Lecturer: Mohammad R. Salavatipour *Scribe: Based on older notes by Sina Khankhajeh*

18.1 Introduction

Euclidean TSP is a subset of **Travelling Salesman Problem** in which distances are on Euclidean plane, i.e. the instances are on $\mathbb{R}^2$. Mathematically speaking, distance of any two vertices $v_i = (x_i, y_i)$ and $v_j = (x_j, y_j)$ is $d(v_i, v_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

We will give a **PTAS** for this problem, i.e. for any $\epsilon > 0$, we will get to a $(1+\epsilon)$-approximation.[1]

The general steps for this matter will be:

- Show that it is sufficient to focus on some special instances having certain properties.
- Use the method of dividing (recursively) the plane into smaller squares.
- Show that there is a $(1+\epsilon) * OPT$ costing tour that does not cross the squares too many times.
- Use dynamic programming.

Several other problems have been subsequently solved using the these ideas such as Steiner Tree.

18.2 Reducing to Nice instances

Our goal is to show that we can have some structural properties about the solution at a small loss (i.e. $(1+\epsilon)$ factor). The number of structural properties we prove are constant; therefore after restricting to such instances we still loose a small ratio in the approximation factor. Our first goal is to show that we can assume the minimum and maximum distance between the points of input is polynomially bounded and all the points have integer co-ordinations on a grid.

Definition 1 ***Nice Instance****: In a nice instance of a Euclidean TSP, minimum distance between points are at least 4, and all the points are integers in* $[0, O(n)]$.

Lemma 1 *We can reduce any Euclidean instance to a nice one at a loss of* $(1+\epsilon)$*-factor.*

Proof. Take a minimal bounding box; say size L. this means there are two vertices with distance at least L and obvious we will have $OPT \geq L$. Take a grid with spacing $\frac{\epsilon L}{2n}$ and move each point to nearest grid point.

[1] The figures and their captions are captured from textbooks of this course [1] [2]

The increase in distances will be at most $\frac{2\epsilon L}{2n}$ for each point. Therefore, the OPT tour will increase by at most $n * \frac{2\epsilon L}{2n} \leq \epsilon * OPT$.

Now, scale things by $\frac{8n}{\epsilon L}$. Now each grid size is at least $\frac{\epsilon L}{2n} * \frac{8n}{\epsilon L} = 4$. So any two points in the scaled version will have distance at least 4.

Note that:

$$\text{Maximum distance (new L)} \ \leq \frac{8n}{\epsilon L} = O(n) \ \ (\epsilon \text{ is fixed})$$

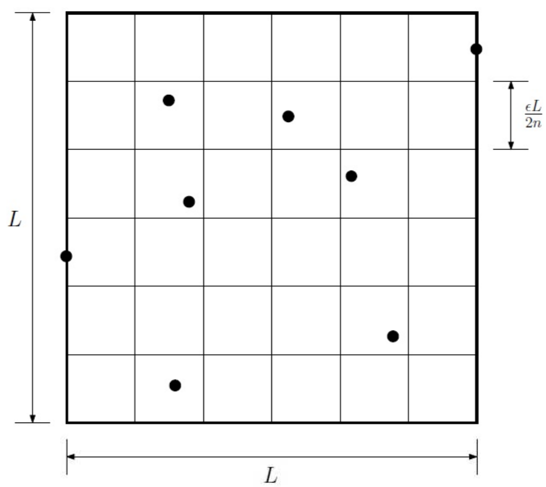

Figure 18.1: An example of the smallest square containing all the points of the instance, and the grid of lines spaced $\frac{\epsilon L}{2n}$ apart

■

18.3 Dissection of bounding box

So far, we can assume that we have a nice instance at a loss of $(1+\epsilon)$-factor. We can assume that:

$$L = 2^k, (\text{and since} L = O(n)) \longrightarrow k = O(\log n)$$

We consider a dissection of bounding box into 4 squares, recursively. This corresponds to a quad-tree such that:
Level Zero(root) corresponds to the bounding box
Level one corresponds to squares of size $\frac{L}{2} \times \frac{L}{2}$
and generally:
Level i corresponds to squares of size $\frac{L}{2^i} \times \frac{L}{2^i}$

We continue until we get to squares of size one. Since our instance is a nice one, there is at most one vertex at each leaf. We define squares having a node inside a **useful squares**. Note that lines dissecting a square of level $i-1$ have level i.

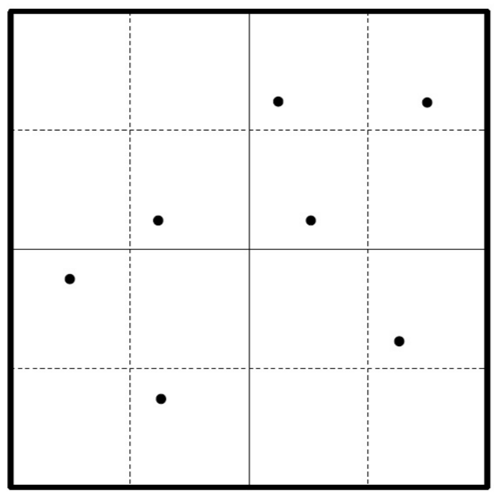

Figure 18.2: An example of a dissection. The level 0 square is outlined in a thick black line; the level 1 squares in thin black lines, and the level 2 squares in dashed lines. The thin black lines are level 1 lines, and the dashed lines are level 2 lines

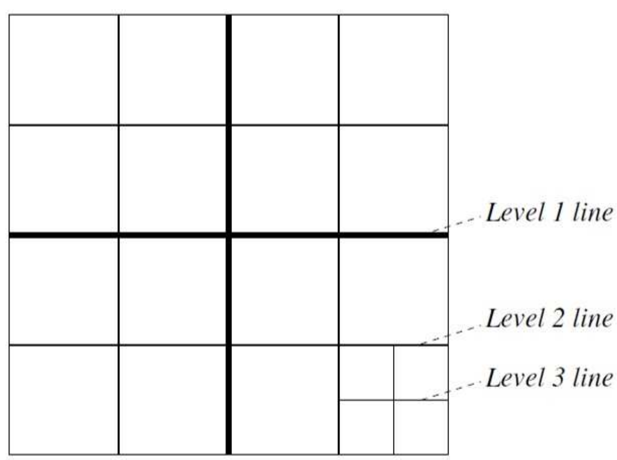

Figure 18.3: Another example of dissection of bounding box

Now, break each line into $m+1$ points called **portals**, that are all equidistant, i.e. on line level i they are $\dfrac{L}{2^i m}$ apart.
Note that portals of larger/higher squares are co-located with smaller ones.
Choose m to be a power of two in $[\frac{k}{\epsilon}, \frac{2k}{\epsilon}]$. So, $m \in O(\frac{\log n}{\epsilon})$.

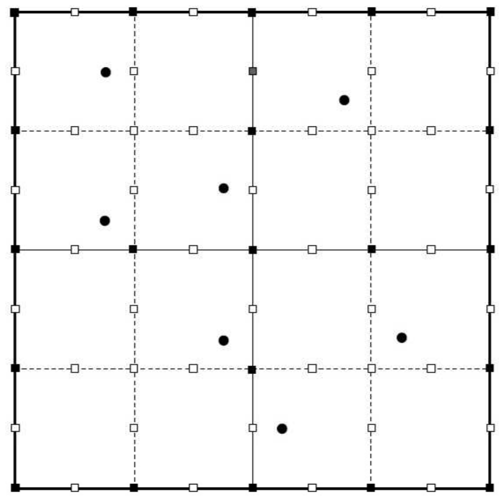

Figure 18.4: Portals added to the squares of the dissection with portal parameter $m = 2$. The black squares are portals for both the level 1 and level 2 squares; the white squares are portals for the level 2 squares.

18.4 Tours with nice features

Definition 2 ***p-tour(portal-respecting tour)***: *a tour that crosses each square only at portal points.*

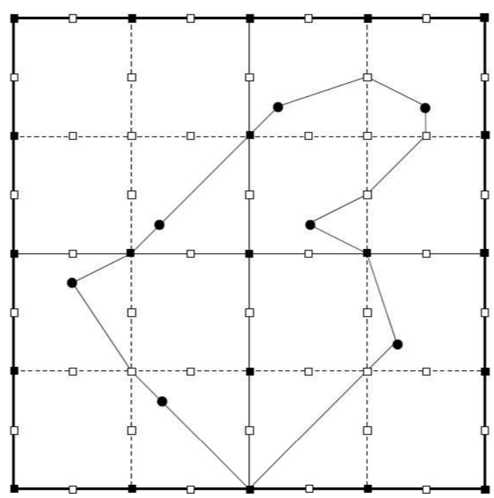

Figure 18.5: A portal-respecting tour(p-tour). Note that only black portals are used to enter or exit the level 1 squares, while the portals on the level 2 squares are added to enter or exit the level 2 squares.

Definition 3 ***2-light p-tour***: *a p-tour that each portal point is crossed at most 2 times.*

Definition 4 ***well-behaved p-tour***: *a p-tour that is non-self-crossing.*

Lemma 2 *if τ is a p-tour that is well-behaved then there is one p-tour that is well-behaved and 2-light of no more cost.*

Proof. By short-cutting and using triangle inequality it is easy to show that we can reduce number of crossings to at most 2.

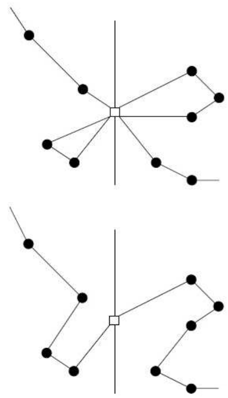

Figure 18.6: Illustration of a short-cutting a portal

■

Lemma 3 *An optimum p-tour that is well-behaved and 2-light can be computed in time* $n^{O(\frac{1}{\epsilon})}$.

Proof. We use dynamic programming. Assume τ is optimum such tour. Consider S to be a square with m portals on each side. τ crosses S at most $8m$ times(each portal at most 2). τ inside S is a collection of at most $8m$ non-intersecting paths.

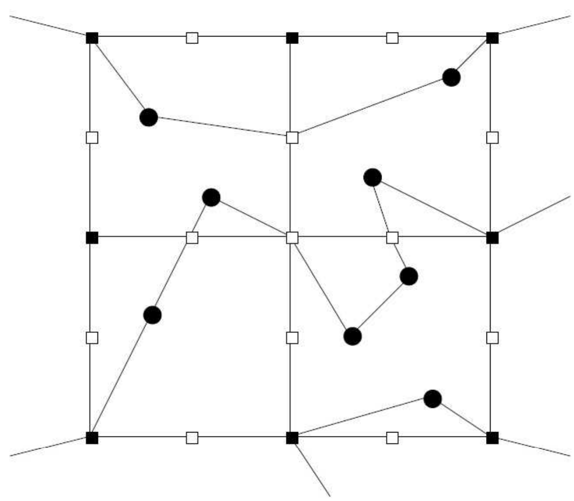

Figure 18.7: Partial p-tour inside a square

In order to find a valid path we know that it must be a non-intersecting path. So if we show the sides of the square S on a single line a valid path will correspond to a valid collection of parenthesis, i.e. for every (there is a unique) at the rest of list of parenthesis. So, the number of valid paths is the $8m^{th}$ number on Catalan numbers which is $2^{O(n)} = n^{O(\frac{1}{\epsilon})}$.

Also, for each portal we can have $0, 1, 2$ crossings. So total number of options are $3^{4m} = n^{O(\frac{1}{\epsilon})}$. Thus there are at most $n^{O(\frac{1}{\epsilon})}$ possible options for how many crossings from each portal and $n^{O(\frac{1}{\epsilon})}$ ways for these portals to be paird. For each such pairing we have a table entry in our DP. We build the DP table in a bottom up manner starting at the lowest level of quad-tree. The base case (for a leaf node) is trivial as there is a small number of portals and at most one point inside the square. For every square S and for every entry of the DP table for this square (for the guess of portals of S as well as the pairing of them) we look at the 4 subproblems corresponding to the 4 children of S and all table entries for those subproblems that are consistent with the guessing on the bigger square portals and consistent with each other; we find a minimum among all subproblems that are consistent with the bigger subproblem/etnry. The number of such entries that we have to check is still $n^{4\times O(1/\epsilon)}$.

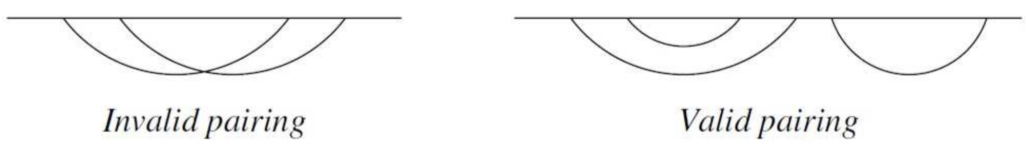

Figure 18.8: An example of invalid and valid pairing

■

18.5 PTAS for Euclidean TSP

So far we have described that we can compute an optimum p-tour 2-light solution for nice instances. We show how the cost of such solution can be bounded away w.r.t. optimum solution if we use randomness in how we do our diessction. For that matter we should first define a random dissection.

Definition 5 *A random dissection is a dissection in which we can choose the origin randomly. It is also called (a,b)-dissection where* $a, b \in [0, \frac{L}{2}]$

The idea is that all the horizontal dissection lines ℓ *we generated are going to have* x *co-ordinate* $+a$ *mod* L *and the vertical lines have* y *co-ordinate* $+b$ *mod* L.

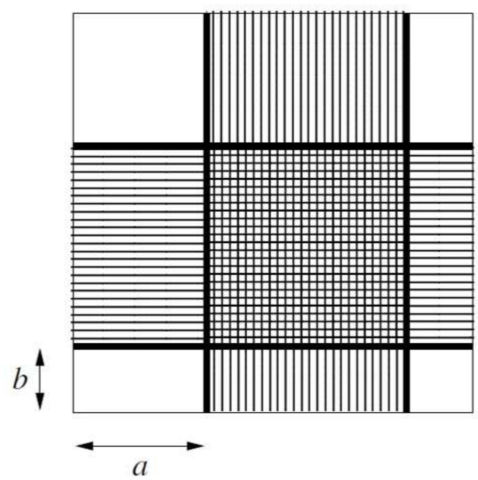

Figure 18.9: A (a,b)-dissection

Theorem 1 *For random (a,b)-dissection, there is a well-behaved 2-light p-tour of cost at most* $(1+\epsilon) * OPT$ *with probability at least* $\frac{1}{2}$.

Proof.

For each line l let $t(l)$ be the number of times the optimum tour crosses l. let $T = \sum_l t(l)$. We claim that $T \leq 2 * OPT$. To prove that, note that for each edge $e = (u, v)$, contribution of e to T will be at most $|x_u - x_u| + |y_u - y_v| + 2$. Cost of e to the optimum is $S = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2} \geq 4$. It is very easy to check that contribution of e to T is at most $2S$.

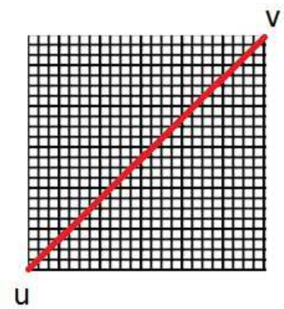

Figure 18.10: u,v and their contribution to T and optimum

Consider any dissection and lets try to make optimum tour τ a well-behaved 2-light p-tour. If τ crosses line l at any point, make it cross at nearest portal.

Because of randomness, we will have:

$$prob[l \text{ is at level i}] = \frac{2^i}{L}$$

So, portal distances on line l are $\frac{L}{2^i m}$.
Therefore, the expected increase in cost by making τ, portal-respecting at intersection with line l is at most:

$$\sum_{i=0}^{k-1} \frac{L}{2^i m} \times \frac{2^i}{L} = \frac{k}{m} \leq \epsilon$$

In above, $\dfrac{L}{2^i m}$ is portal distances and $\dfrac{2^i}{L}$ is the probablity of line l being of level i.

Summing up over all lines (i.e. all crossings in T) we get an upper bound for increase in cost of $\epsilon * 2OPqT$.

■

So we have a $(1+\epsilon^{'})$-factor here and we had a $(1+\epsilon^{''})$-factor at the beginning to turn our instance to a nice instance. Overall, we will still have a $(1+\epsilon)$-factor for our general Euclidean problem which is the PTAS we were looking for.

Actually this technique is operable on d-dimensional Euclidean TSP and we can get a PTAS in that problem as well.

References

[1] V.V. Vazirani. *Approximation algorithms.* Springer Verlag, 2001.

[2] D.P. Williamson and D.B. Shmoys. *The design of approximation algorithms.* Cambridge Univ Pr, 2011.