COT5405 – Analysis of Algorithms – Spring 2020

Survey Paper

Polynomial Time Approximation Scheme for Euclidean TSP

Suprith Reddy Gurudu

UFID: 9961-2134

Introduction

The Traveling Salesman Problem (TSP) is one of the popular NP-complete problems. Given a weighted graph G of n vertices, we need to compute a cycle with minimum cost that visits each vertex of G only once. Our concern is primarily focused on approximate solutions (mostly for special cases of TSPs) because computing optimal/exact solution is too expensive (i.e., exponential time complexity assuming $P \neq NP$). Especially, we are interested in a Polynomial Time Approximation Scheme (PTAS) which means the computation is bounded by polynomial (in terms of n) running time and the total cost of computed tour is not more than (1+ϵ) times *OPT* where ϵ is a fixed parameter greater than zero and *OPT* stands for total optimal tour cost. Few special cases of Traveling Salesman Problem include Metric TSP (edge weights satisfy *triangle inequality* – $d(x, y) + d(y, z) \geq d(x, z)$), Euclidean TSP (Euclidean metric distance - itself a special case of Metric TSP), Rectilinear TSP (Manhattan metric distance) and Asymmetric TSP (directed edge weights). Assuming distances between vertices are metric, there are some approximation algorithms such as two-factor approximation algorithm[4] based on minimum spanning tree doubling and Eulerian path/circuit runs in $O(m + \log n)$ and $\frac{3}{2}$ - factor approximation algorithm presented by Christofides[7] that runs in $O(n^3)$. We cannot get an approximating factor less than $\frac{123}{122}$ on our assumption $P \neq NP$[3].

Euclidean TSP: For a fixed dimension d, Euclidean distance cost function *distance(x, y)* = $\sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$ where x, y ∈ set of n points in $\mathbb{R}^d$ space.

To make it less complex, we focus on d = 2, i.e., n points in an $\mathbb{R}^2$ space or a simple plane. We use Dynamic Programming strategy to describe a PTAS for the Euclidean TSP. Before applying Dynamic Programming, we initially need to simplify the input and limit the solution space.

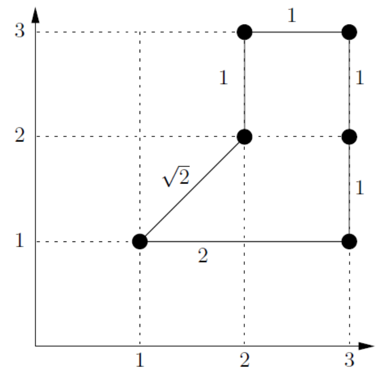


Figure 1. A Euclidean TSP example with tour cost of $6 + \sqrt{2}$

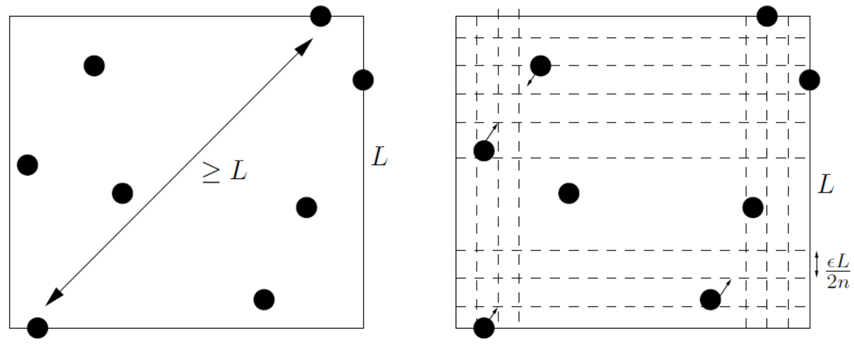
Figure 1[5] shows an instance of a Euclidean TSP. In the above example the length of the tour is clearly irrational. Since we do not know how to compute exact solution for square roots and can only find an approximate value, we do not know whether Euclidean TSP is in NP even though it is NP-hard. We will use following steps to obtain a PTAS [1] for this problem: rounding off the instance, dividing the space into squares and applying dynamic programming strategy to these portions.

Preliminaries

Bounding Box

A bounding box is defined with the smallest square containing all points (i.e., n). Let L be the length of each edge equal to $4n^2$ and we can do this by extending all distances by a specific factor. Apparently, there is no change in the optimal tour. Assuming, without the loss of generality, $n=2^p$, implies that $L = 2^k$, where p and k are some positive integers. Therefore, $k = 2 + 2\log_2 n = O(\log_2 n)$.

Moreover, we move every vertex of G to the closest grid-point as shown in the Figure 2 so every vertex will be an integer point and it is bounded. As two points (at least) are located at opposite edges of the box, a lower bound for OPT is two times L, if not the box would be smaller. We can clearly see (from figure 2) that the maximum distance from any arbitrary point to the closest grid-point is $\frac{1}{\sqrt{2}}$ so the absolute error per vertex point is bounded by $\sqrt{2}$. As the distance cannot be more than twice the maximum distance, the total absolute error is bounded by $\sqrt{2}n$. Therefore, we get a bounded relative error $r \leq \frac{Total\ Absolute\ Error}{Optimal\ Tour\ \text{Cost}} = \frac{\sqrt{2}n}{OPT} \leq \frac{\sqrt{2}n}{2L} = \frac{\sqrt{2}n}{8n^2} = \frac{1}{4\sqrt{2}n}$. We can deduce that for all $\varepsilon > 0$ there exists n_0, for all $n \geq n_0$ such that $r \leq \varepsilon$. Hence, by proper adjustment of ε, we can compensate the relative error.



(a) *(b)*

Figure 2. (a) Bounding box illustration of length L and (b) Scaled box with fine grid

Dissection

A basic dissection is dividing a bounding box into levels of squares. Two level one lines into four level one squares of side length $\frac{L}{2}$ and four level two lines into another four level two squares of side length $\frac{L}{2^2}$ and so on until unit length squares are obtained as shown in the Figure 3. In general case, a level *i* square has side length of $\frac{L}{2^i}$.

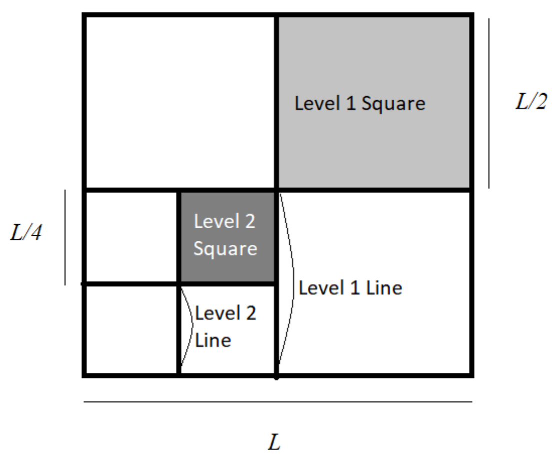


Figure 3. Dissection of the bounding box

Quad (4-ary) Tree

A basic dissection can be represented as a quad tree as shown in the Figure 4 with height of the tree *k* and the number of tree nodes equals to $\frac{4^{k+1}-1}{4-1} = \frac{4^{(2+2\log_2 n)+1}-1}{3} = O(n^4)$.

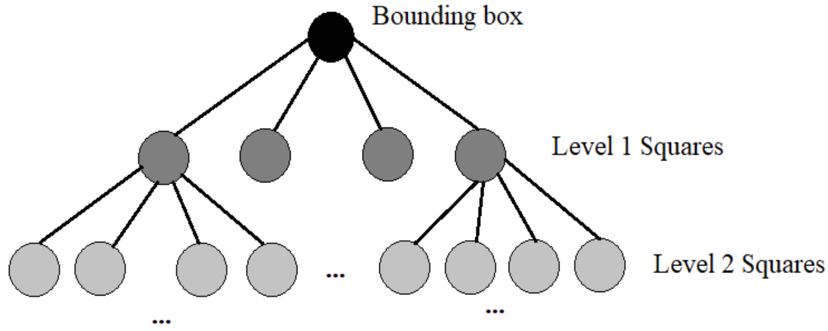


Figure 4. Quad Tree Illustration for the bounding box

Portals

The certain points at which lines of dissection can only be crossed are called portals. Each corner of the square and additional *m-1* divisions of an edge are the portals which amounts to a total of *4m* portals for each square as shown in the Figure 5. The *m* parameter must be in between (closed interval) $\frac{k}{\varepsilon}$ and $\frac{2k}{\varepsilon}$ and $m = 2^c$ where *c* is some positive integer. Thus, portal of level *i* square is also a portal for level *i+1* and distance between any two portals of a level *i* square is $\frac{L}{2^i m}$.

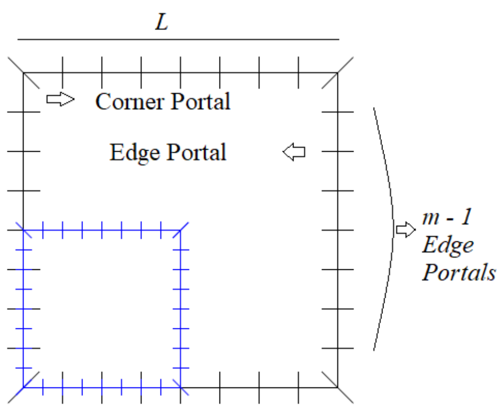


Figure 5. An illustration for the portals

Approximation Technique

ε - nice instance

Definition 1: ε – nice is an instance of Euclidean TSP if it follows two conditions[5]:

(i) The integral co-ordinates of every point must lie in the interval $[0, O\left(\frac{n}{\varepsilon}\right)]^2$.

(ii) The distances between any two distinct points must be at least 4.

As already mentioned above in the bounding box, we can translate and scale the input instance with same factor by preserving the approximation ratio as shown in the Figure 2(b). Specifically, we must consider the smallest box around the points of the instance and let L be the longest side length of the box. We can translate and scale in such a way that L becomes $\lceil 8n/\varepsilon \rceil$.

Lemma 1: Let P be an instance of Euclidean TSP[5]. Let OPT_P denote the length of the optimal tour in P. We can transform P into an ε-nice instance P' such that $OPT_{P'} \leq (1 + \varepsilon)OPT_P$.

Proof

As shown in the Figure 2(a), consider the smallest bounding box around the points of instance P. The longest side has length L and we know that there exists two points in the smallest box on opposite sides with at least distance of L. Therefore, the size of the optimal tour is at least L (i.e., $OPT_P \geq L$, indeed, $OPT_P \geq 2L$ because tour has to be back and forth between two points). To acquire an instance P' we can draw a fine grid of spacing $\frac{\varepsilon L}{2n}$ into the bounding box and each can be mapped to its nearest grid-point. Nonetheless, some points in P might be the same point in P' as well. Certainly, all points are integer co-ordinates in a respective range because $L = \lceil 8n/\varepsilon \rceil \in O\left(\frac{n}{\varepsilon}\right)$. Moreover, the grid spacing is $\frac{\varepsilon L}{2n} \geq \frac{\varepsilon}{2n} \frac{8n}{\varepsilon} = 4$. Hence, the distance between any two distinct points in P' is at least four and it implies P' is ε-nice.

We need to prove that optimal tours in P and P' differ by at most a factor of $(1 + \varepsilon)$. Consider the optimal tour in P and map the points of P to the points in P' by translating every point at most of $\frac{\varepsilon L}{2n}$. This implies that every edge is changed by at most $\frac{\varepsilon L}{n}$ (zero if two points in P and P' are same) between two points and the cost change is at most εL. We got a tour with at most $OPT_P + \varepsilon L$ for P' and we know that $L \leq OPT_P$, by combining these two we get

$$\boxed{OPT_{P'} \leq OPT_P + \varepsilon L \leq (1 + \varepsilon)OPT_P} \quad \blacksquare$$

As we already mentioned that we need to restrict our tour with respect to the basic dissection:

Definition 2: A tour T on the n points is well behaved with respect to the basic dissection if T is any subset of portals.

Our primary aim is to find a well-behaved optimal tour T and show that || T || (length of the tour) is less than or equal to $(1 + \varepsilon)OPT$.

Definition 3: A tour T that is well behaved with respect to the basic dissection has limited crossings if T visits each portal at most twice and is non-self-interesting.

Lemma 2: Let T be the well-behaved tour with respect to the basic dissection. This implies there is a T' tour that is well-behaved with respect to the basic dissection and has limited crossings so that $\|T'\| \leq \|T\|$.

Proof (Correctness)

Due to the triangle inequality property of Euclidean TSP, shortcutting the distance of the tour by removing intersections will not increase the tour length. Hence this lemma holds true. From lemma 1, we can say that $\|T'\| \leq (1+\varepsilon)\|T\|$. Therefore, with limited crossings, T' is short enough so that $\|T'\| \leq \|T\|$.

Thus, we must do the following two steps to get *(1+ε)* factor approximation:

(a) show that there is a well-behaved tour T that is short enough and
(b) find an optimal tour T with limited crossings using dynamic programming

Dynamic Programming

Each portal can only be crossed at most twice (0, 1 or 2). We have 3^{4m} ($= n^{O\left(\frac{1}{\varepsilon}\right)}$) possibilities because the number of portals of a square is *4m*. Moreover, we claim that invalid pairings shown in Figure 6(a) are not allowed i.e., no self-intersection is allowed, and only valid ones shown in Figure 6(b) are allowed. A valid pair associates to the balanced parenthesis as shown in the Figure 6(c). When *2r* portals are used, the number of valid pairings will be equal to the r^{th} Catalan number represented as *C(r)*. This can be bounded by 2^{2r} since there are two possibilities (left and right) for each of the *2r* parenthesis/portals. Moreover, we know that each portal is visited at most twice, thus, $2r \leq 2(4m) = 8m$. Therefore, the number of valid pairings is bounded by $n^{O\left(\frac{1}{\varepsilon}\right)}$ because $C(r) \leq 2^{2r} \leq 2^{8m} = n^{O\left(\frac{1}{\varepsilon}\right)}$. By combining both results, we get a total possibilities as $n^{O\left(\frac{1}{\varepsilon}\right)}$ (for portal usage) times $n^{O\left(\frac{1}{\varepsilon}\right)}$ (for valid pairings) equals to $n^{O\left(\frac{1}{\varepsilon}\right)}$.

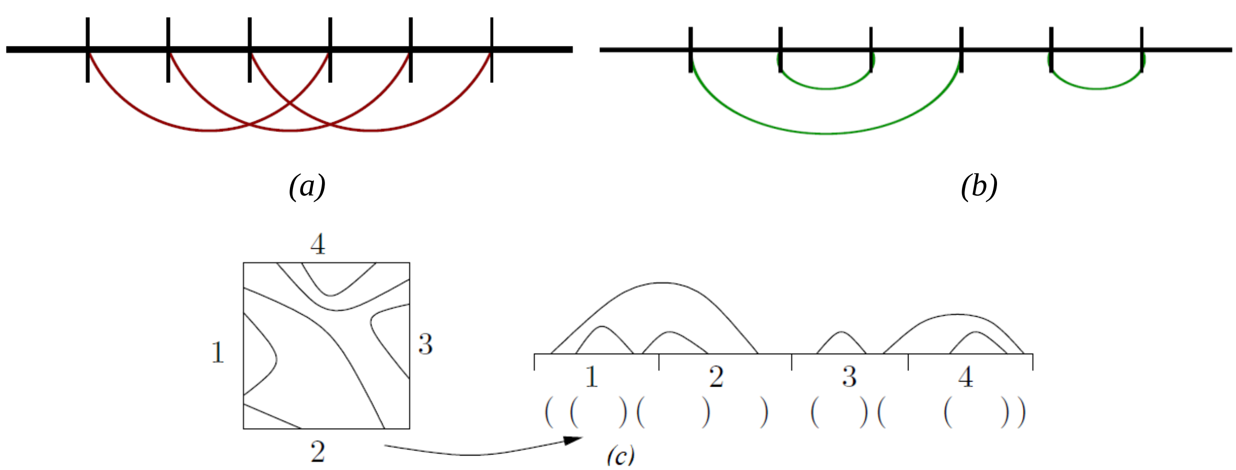
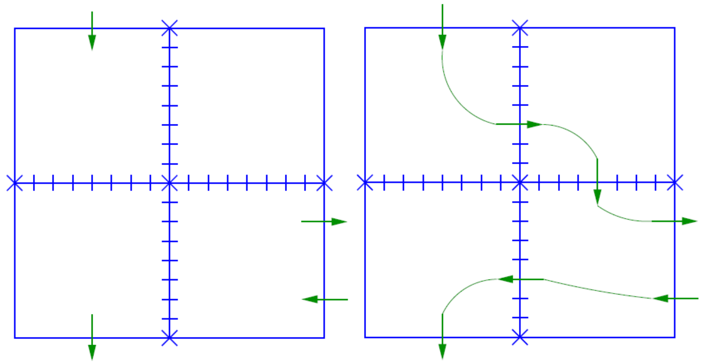


Figure 6. (a)Invalid portals with intersections (b)Valid portals with no intersections (c)Parenthesis representation of portals

The algorithm uses 2-D dynamic programming table and fills up in the bottom-up approach. The number of columns is equal to the number of rows which in turn equals to the number of nodes in the quad tree (i.e., $O(n^4)$)represented above. The valid visits are represented using the row cells, so the number of rows is bounded by $n^{O\left(\frac{1}{\varepsilon}\right)}$. Thus, the table size equals to $O(n^4).n^{O\left(\frac{1}{\varepsilon}\right)} = n^{O\left(\frac{1}{\varepsilon}\right)}$.

Each valid visit shown in the Figure 7(a) corresponds to an entry in the row of the dynamic programming table. We know that each square has $n^{O\left(\frac{1}{\varepsilon}\right)}$ possibilities of portal crossings. Since, we know each square's possible entries, summing up all the appropriate valid entries as shown in the Figure 7(b) and finding the minimum of all the results produces the optimal length desired. Therefore, the total expense of this algorithm will be table size times each entry expense which leads to $n^{O\left(\frac{1}{\varepsilon}\right)} . n^{O\left(\frac{1}{\varepsilon}\right)} = n^{O\left(\frac{1}{\varepsilon}\right)}$.

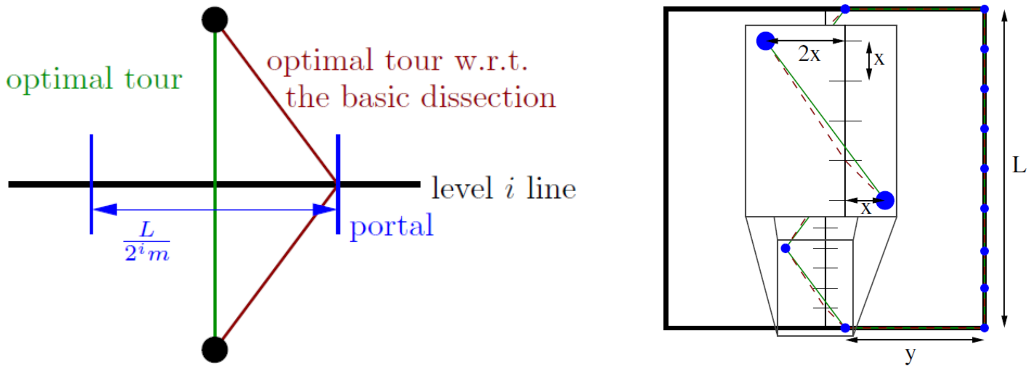


(a) *(b)*

Figure 7. (a)Usage of portals via green arrows (b)Valid entries through portals

Losses through Basic Dissection

The Dynamic programming approach is possible because crossing lines arbitrarily is not allowed, but this leads to a problem of increasing the length of the tour as shown in the Figure 8(a).



(a) *(b)*

Figure 8. Tour length illustration with two different approaches.

As shown in the Figure 8(b), the length of the optimal tour is represented as green line which is equal to $OPT = \frac{\sqrt{3^2+4^2}}{4}L + 2y + L$ and the length of the optimal tour with respect to the basic dissection is represented as red line which is equal to $OPT_A = \frac{\sqrt{2}+\sqrt{2^2+3^2}}{4}L + 2y + L$. We know that $y \leq \frac{L}{2}$, we get the ratio between OPT_A and OPT as

$$\boxed{\frac{OPT_A}{OPT} \geq \frac{1.2549L + 2.\frac{L}{2} + L}{1.25L + 2.\frac{L}{2} + L} = 1.0015}$$

So, the algorithm fails if we choose the parameter $\varepsilon < 0.0015$ and this problem can be solved by randomizing the algorithm and generalizing the basic dissection concept.

Conclusion

When *d* = *n* – *1*, the above problem becomes Max-SNP-hard[6]. Given a constant $\varepsilon_1 > 0$, the *(n-1)*-dimensional Euclidean TSP cannot be approximated within a factor of (1+ε_1) in polynomial time unless *P* = *NP*. Under randomized reductions, we can approximate the *c log n*-dimensional Euclidean TSP within (1+ε_2), where *c* and $\varepsilon_2 > 0$ are some constants. This contradicts the result produced by Arora[1].

References

[1] S Arora. Polynomial Time Approximation Schemes for Euclidean TSP and other Geometric Problems: *https://dl.acm.org/doi/10.1145/290179.290180*

[2] Dominik Schultes. Euclidean Traveling Salesman Problem: *http://algo2.iti.kit.edu/schultes/lebenslauf/publications/euclTSPsummary.pdf*

[3] M Karpinski. M Lampis. R Schmied. New Inapproximability Bounds for TSP: *https://arxiv.org/pdf/1303.6437v1.pdf*

[4] Introduction To Algorithms, 3rd Edition: C.L.R.S

[5] Ola Svensson. Approximation Algorithms and Hardness of Approximation: *https://theory.epfl.ch/osven/courses/Approx13/Notes/lecture4-5.pdf*

[6] L Trevisan. On the Approximability of the Multi-dimensional Euclidean TSP: *https://eccc.weizmann.ac.il/report/1996/046/*

[7] N Christofides. Worst-case analysis of a new heuristic for the TSP. Technical report, CMU, 1976