Saarland
Informatics Campus

Script of Lecture 3, Approximation Algorithms
Summer term 2017

Tobias Mömke, Hang Zhou
`http://www-cc.cs.uni-saarland.de/course/61/`

Written by Tobias Mömke

1 Euclidean TSP

For a given number k, we consider the Euclidean space $\mathbb{R}^k$. An instance of Euclidean TSP is a complete graph G with its vertices V points (vectors) in an Euclidean space.

Lemma 1. *Every norm $\|\cdot\|$ has an induced metric determined by $w(x,y) = \|x-y\|$.*

The lemma follows from a simple application of the norm properties (but the proof is not part of this course).

In Euclidean TSP, the distance between two vertices x, y is the induced metric of the Euclidean norm, i. e.,

$$w(x,y) := \|x-y\| = \sqrt{\sum_i (x_i - y_i)^2}\,.$$

Intuitively, the weight is the length of the straight line between two points.

For simplicity, in the following we only consider the Euclidean plane $\mathbb{R}^2$.

We observe that adding additional points from $\mathbb{R}^2$ can only increase the length of the computed cycle: since w is a metric we can skip these point later on (i. e., we use shortcuts). The same is true if we introduce multiple copies of one point.

2 Arora's Algorithm

We show that there is a PTAS for Euclidean TSP in $\mathbb{R}^2$. The ideas of this algorithm have inspired a large number of algorithms for other problems. The main part of the algorithm is to transform an (unknown) optimal solution into an almost optimal solution with good properties. Afterwords we show that we can compute such a solution.

2.1 Bounding Box

We first scale the instance. If we see two points x, y as a vector, we can replace them by αx and αy. By the properties of a norm, $w(\alpha x, \alpha y) = \|\alpha x - \alpha y\| = \alpha\|x-y\|$. Therefore all distances are scaled by α.

We choose an $\alpha > 0$ such that all vertices fit exactly into an axis-parallel $n^2 \times n^2$ square (where $n = |V|$), i. e., all vertices fit into the square and there are two vertices that lie on opposite boundaries of the square. We assume $n = 2^{k'}$ for an integer k' (add up to $|V|$ new vertices until a power of 2 is reached). Then $L := n^2 = 2^k$ for $k = 2\log_2 n$.

Within the $L \times L$ box, we move each vertex to the closest center of an integer 1×1 square (break ties arbitrarily). This step is called perturbation.

Lemma 2. *A $(1+O(\epsilon))$-approximation of the scaled and perturbed instance determines a $(1+O(\epsilon))$-approximation of the original instance.*

The proof is left as an exercises.

2.2 Quadtree Dissection

We divide the $L \times L$ box into four $L/2 \times L/2$ boxes and continue subdividing squares into four smaller squares. These separations determine a quadtree where the root (level 0) is the $L \times L$ square. The leaves correspond to 1×1 squares: the dissection stops after k steps.

Level i square: square introduced at level i. Level i lines: lines of length L composed of the new boundaries introduced in level i of the dissection (i. e., inner boundaries of $L/2^i \times L/2^i$ squares).

2.3 Portals

Let $m := \lceil k/\epsilon \rceil$ (and thus $m = O((\log n)/\epsilon)$). For each i, we place m equidistant *portals* at each side of a level i square that does not belong to a previous square (of level $i' < i$).

A tour (i. e., solution) is *portal respecting*, if it crosses boundaries of the squares only through portals. A tour is k''-light if it crosses no side of a dissection square more than k'' times.

Lemma 3. *Let τ be a portal respecting tour with respect to the quadtree dissection. Then there is a $2m$-light portal respecting tour τ' with $w(\tau') \le w(\tau)$ such that τ' crosses each portal at most twice.*

Proof. Let p be a portal that is crossed more than twice. Let us assume that τ has an orientation (that is, we follow the tour in a given direction). Then p is crossed at least twice from the same side. We "cut" the tour at both crossings and close them at the two sides of the portal. Checking the correctness is left as an exercise. □

Additionally, we would like to avoid self-intersections. Transforming a tour into another tour without self-intersections can be done similarly to avoiding multiple portal crossings and is left as an exercise.

After the uncrossing, within each square the tour has a parentheses structure: the paths between two distinct boundaries behave as if they were parenthesis. Each square has up to $8m$ portal crossings and thus up to $4m$ parenthesis pairs. The number of possible arrangements is determined by the Catalan numbers. The total number can be bounded by $2^{4m} \in n^{O(1)}$.

2.4 Shifted Quadtree

Unfortunately, a portal respecting solution can be much longer than an optimal solution that is not portal respecting.

In order to solve this problem, we introduce a shift: we choose two numbers a, b such that $0 \le a, b < L$. We move each vertical line at position x to position $(x + a) \mod L$ and each horizontal line at position y to $(y + b) \mod L$.

We obtain the same number of squares and lines, but some of the squares are wrapped around the boundary.

In the following, we consider the best of the L^2 choices of pairs (a, b).

2.5 Patching Lemma

The core of the analysis is to show that there is a shifted quadtree such that there is an almost optimal portal respecting solution.

Lemma 4. *Let τ be an optimal TSP tour. There is a shift (a, b) such that there is a portal respecting tour τ' of length at most $(1 + 2\epsilon)w(\tau)$.*

Proof. We first observe that τ crosses at most $2 \cdot w(\tau)$ grid lines. (This is left as an exercise.) We show that on average, each crossing increases the length of the tour by at most ϵ. Then the overall increase is at most $2\epsilon w(\tau)$, as claimed.

We choose a and b uniformly at random from their ranges and show that the claim holds in expectation.

The line crossings of τ are fixed (at each integer coordinate), but the level depends on the shift. For a line crossing, the probability that the crossed line l is of level i is $2^i/(2L)$ since there are 2^i lines of level i and the total number of lines is $2L$. There is one additional line of level 0. The closest portal of a level i line is at most $\frac{1}{2}L/(2^i m)$ away from the crossing. The detour to the portal costs $L/(2^i m)$.

In total:

$$\frac{L}{m} \cdot \frac{1}{2L} + \sum_{i=0}^{k} \frac{L}{2^i m} \frac{2^i}{2L} = (k+2)/(2m) \leq \epsilon \,.$$

The claim follows by linearity of expectation. Observe that trying all possible choices of a and b cannot be worse than choosing them randomly. □

Corollary 1. *Let τ be an optimal solution. Then there is a shifted quadtree dissection with a solution τ' that is portal respecting, has at most 2 crossings per portal, does not have self-intersections, and $w(\tau') \leq (1 + O(\epsilon))w(\tau)$.*

2.6 The Dynamic Program

We compute a solution for each DP cell that stores the following information.

- The shifting parameters a and b;
- a level i;
- a level i square of the quadtree dissection shifted by a,b;
- the pairs of portals such that there is a path connecting them.

The algorithm first computes all level k cells: these cells contain only one point (at the middle of the cell). Observe that it is sufficient if one path visits the center (or zero if there is no vertex inside the square). We can compute an exact solution by trying all possibilities.

If all levels greater than i are computed, we compute the value of a level i cell C as follows. We try each combination of 4 level $i + 1$ cells that cover C. For each selection we check that the number of portal crossings of neighboring portals fits. Then we check if the composition of paths (i. e., pairs of portals) correspond to the paths of C. Observe that the check can be done efficiently since we do not allow crossings. If everything fits, the four sub-cells are *compatible* to C.

The value of C is the smallest sum of cell values over all selection of compatible sub-cells.

Theorem 1. *The algorithm above is a PTAS for the two-dimensional Euclidean TSP.*

Proof. The running time is polynomial: there are only polynomially many choices of a, b, i and we have seen that the number of portal pairs to consider is polynomial (Catalan numbers, $O((\log n)/\epsilon)$ portals).

Computing the leaves only requires to take direct connections between the portals (straight lines) and choosing one line that visits the center.

Computing the non-leaf cells considers each combination of four other cells, which is polynomial since $(n^{O(1)})^4 = n^{4O(1)} = n^{O(1)}$.

The feasibility of the solution follows directly from the construction.

Approximation ratio: By Corollary 1 there is a short light portal-respecting tour τ' without crossings. For each cell, we consider all segments of such a tour. Since we keep the shortest solution for each square, we compute a solution as least as good as τ'. The claim follows. □

3 Outlook

The algorithm holds almost unchanged for the Steiner tree problem.

The key step of the algorithm is the patching lemma. The algorithm as presented here crucially requires the dimension to be at most 2.

There is a more involved version of the patching lemma that can handle additional dimensions: It is possible to show that we can ensure 2-light tours, i. e., the *total* number of crossings at each side of a square is at most 2 (instead of $2 \log n$).

There are recent results and open research questions related to quadtree dissections.

Example: capacitated vehicle routing.

We want to compute a TSP solution, but there is a special vertex called the origin. The instance has an additional parameter κ. After visiting at most κ vertices, the tour has to visit the origin again.

This is motivated for instance by packet delivery: a truck has a limited storage capacity (κ), and after delivering κ items, it has to be refilled at the origin.

There is a QPTAS for capacitated vehicle routing (based on quadtree dissections) but not a PTAS.