Arora's PTAS for the Euclidean TSP[1]

R. Inkulu
http://www.iitg.ac.in/rinkulu/

[1] most of the figures in this presentation are from the references mentioned

problem description

Given a set P of points in Euclidean plane, find a tour of minimum cost that visits all the points of P.

- known to be NP-hard
- objective of this talk: a polynomial-time approximation scheme to compute a tour whose expected length is $(1+\epsilon) \cdot OPT$ in $O(n(\frac{\lg n}{\epsilon})^{O(\frac{1}{\epsilon})})$ time

problem description

Given a set P of points in Euclidean plane, find a tour of minimum cost that visits all the points of P.

- known to be NP-hard
- objective of this talk: a polynomial-time approximation scheme to compute a tour whose expected length is $(1+\epsilon) \cdot OPT$ in $O(n(\frac{\lg n}{\epsilon})^{O(\frac{1}{\epsilon})})$ time

for the convenience, we assume the following:

$\frac{1}{n} < \epsilon < 1$

P is contained in $[\frac{1}{2}, \frac{1}{2}] \times [1, 1]$

$diam(P) \geq \frac{1}{4}$

round the input points to grid points

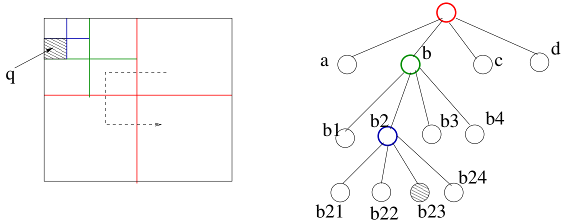
- set up a grid $G = \{\frac{1}{n\lceil\frac{32}{\epsilon}\rceil}(i,j) \mid i,j \text{ are integers}\}$

round the input points to grid points

- set up a grid $G = \{\frac{1}{n\lceil \frac{32}{\epsilon} \rceil}(i,j) \mid i,j \text{ are integers}\}$
- for every point $p \in P$, snap (round) p to an arbitrary corner (grid point) of the grid cell to which p belongs

- let Q be the resulting point set
- reducing P to Q results in quadtree of Q to have logarithmic depth

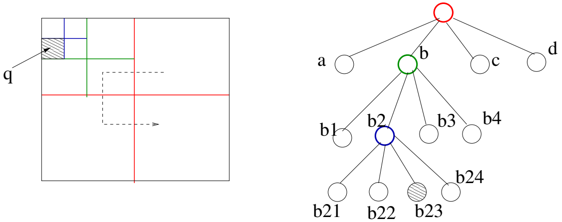
setting up a quadtree over Q



- consider the unit square with its southwest corner at $(\frac{1}{2}, \frac{1}{2})$[2] as the square of the root node

[2] later, this corner will be fixed at a point randomly chosen from the square with the southwest and northeast corners respectively at $(0, 0)$ and $(\frac{1}{2}, \frac{1}{2})$

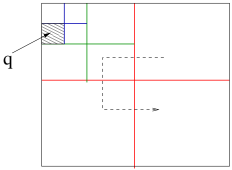
setting up a quadtree over Q

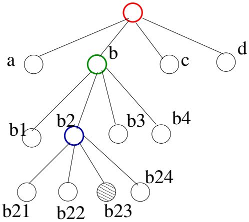


- consider the unit square with its southwest corner at $(\frac{1}{2}, \frac{1}{2})^2$ as the square of the root node
- since the minimal distance between any pair of points of Q is at least $\frac{1}{n\lceil \frac{32}{\epsilon} \rceil}$, it suffices to have the sidelength of square corresponding to any leaf node to be $\frac{1}{n\lceil \frac{32}{\epsilon} \rceil}$

[2] later, this corner will be fixed at a point randomly chosen from the square with the southwest and northeast corners respectively at $(0, 0)$ and $(\frac{1}{2}, \frac{1}{2})$

setting up a quadtree over Q



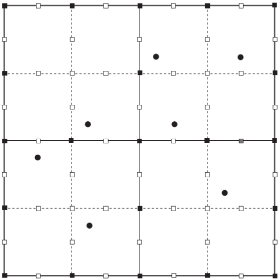


- consider the unit square with its southwest corner at $(\frac{1}{2}, \frac{1}{2})^2$ as the square of the root node
- since the minimal distance between any pair of points of Q is at least $\frac{1}{n\lceil\frac{32}{\epsilon}\rceil}$, it suffices to have the sidelength of square corresponding to any leaf node to be $\frac{1}{n\lceil\frac{32}{\epsilon}\rceil}$
- height H of the quadtree is $O(\lg n)$; letting no node of the quadtree is empty, the number of nodes is $O(n \lg n)$

  level of root is 0; for every $0 \leq i \leq H$, grid corresponding to level i is denoted by G^i, and the length of any edge of G^i is $\frac{1}{2^i}$

[2] later, this corner will be fixed at a point randomly chosen from the square with the southwest and northeast corners respectively at $(0,0)$ and $(\frac{1}{2}, \frac{1}{2})$

portals along the quadtree square sides

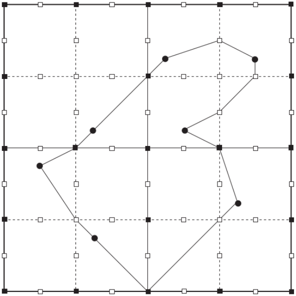


portals in black are from G^1 and G^2 and portals in white are from G^2

- for every i, along each side of each quadtree square of G^i, introduce equi-spaced $m = \frac{20H}{\epsilon}(= O(\frac{\lg n}{\epsilon}))$ *portals*;
- in addition, for each such square, introduce one portal at each of its corners.

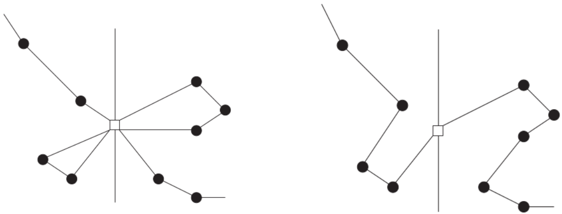
- by choosing $m+1$ as a power of two, each portal on the sides of any square of G^{i-1} coincides with a portal introduced for squares in G^i

requisite characteristics of a constrained tour



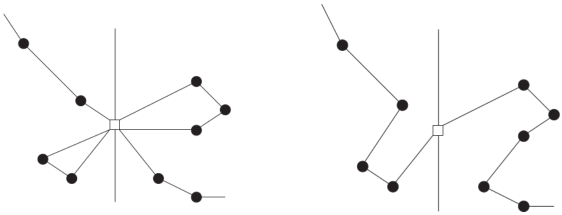
- tour is permitted to cross an edge *e* of a quadree square only via portals on *e*

requisite characteristics of a constrained tour (cont)



- no portal at any level can be used more than twice: once to enter a square and once to exit a square

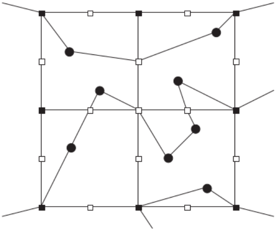
requisite characteristics of a constrained tour (cont)



- no portal at any level can be used more than twice: once to enter a square and once to exit a square
- for $r = \frac{90}{\epsilon} = O(\frac{1}{\epsilon})$, tour needs to be *r-light* with respect to any edge of any square in any level of the quadtree, that is, tour can use at most r portals located on any side of any quadtree square

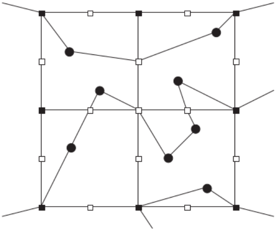
is it feasible to have a tour meeting these constraints while being a good approximation to an optimal tour?

nested parenthesis structure of portal-to-portal paths



- portal-to-portal paths follow parentization; hence, one can try all settings of parenthesis, translate these into possible layouts of paths and discard the ones that have intersecting paths

nested parenthesis structure of portal-to-portal paths



- portal-to-portal paths follow parentization; hence, one can try all settings of parenthesis, translate these into possible layouts of paths and discard the ones that have intersecting paths

- therefore, the number of layouts of k portal-portal paths in a grid square is upper bounded by the k^{th} Catalan number, which is $\frac{1}{k+1}\binom{2k}{k} = O(2^{2k})$

valid solutions of any grid square

- for every 0 to $4r$ portal combination of $4m+4$ portals on the boundary, considering that each chosen portal can be used at most twice, for every permutation p of the chosen portals, alternately mark portals in p with enter-exit

- leading to number of subproblems $\leq \sum_{i=0}^{8r} \binom{2(4m+4)}{i} i! = (\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})}$

valid solutions of any grid square

- for every 0 to $4r$ portal combination of $4m + 4$ portals on the boundary, considering that each chosen portal can be used at most twice, for every permutation p of the chosen portals, alternately mark portals in p with enter-exit
  - leading to number of subproblems $\leq \sum_{i=0}^{8r} \binom{2(4m+4)}{i} i! = (\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})}$
- for every square R in G^i, find a minimum cost collection of non-intersecting subpaths that intersect R which respect portals of R and cover points of P in R

valid solutions of any grid square

- for every 0 to $4r$ portal combination of $4m+4$ portals on the boundary, considering that each chosen portal can be used at most twice, for every permutation p of the chosen portals, alternately mark portals in p with enter-exit
  - leading to number of subproblems $\leq \sum_{i=0}^{8r} \binom{2(4m+4)}{i} i! = (\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})}$
- for every square R in G^i, find a minimum cost collection of non-intersecting subpaths that intersect R which respect portals of R and cover points of P in R

— to achieve efficiency, build these in bottom-to-top fashion from leaf nodes to the root of the quadtree while memoizing with DP

dynamic programming over squares of quadtree grid

- interaction between problems is according to the organization of quadtree nodes

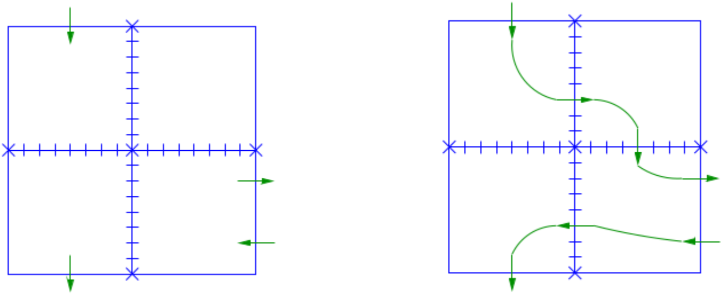
dynamic programming over squares of quadtree grid

- interaction between problems is according to the organization of quadtree nodes
- memoize the subproblem solutions

dynamic programming over squares of quadtree grid

- interaction between problems is according to the organization of quadtree nodes
- memoize the subproblem solutions
- base case: if a square has $O(\frac{1}{\epsilon})$ points of Q, then solve that subproblem by brute force

DP: assemble subtours from children



- at every non-leaf node, for every chosen portal permutation, save these entries in the DP table: portal permutations of each child node which together caused the minimum cost subtour at the parent

time complexity

- number of subproblems to pursue at each quadtree node $\leq \sum_{i=0}^{8r} \binom{2(4m+4)}{i} i! = (\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})})$

time complexity

- number of subproblems to pursue at each quadtree node $\leq \sum_{i=0}^{8r} \binom{2(4m+4)}{i} i! = (\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})}$
- subproblem at each quadtree node, ignoring the recursive subproblems, can be solved in $((\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})})^{O(1)}$ time
  - assembling: each subproblem solution of 1^{st} child with each ... with each subproblem solution of 4^{th} child
  - while verifying whether the assembled subtour is according to enter-exit constraints and non-intersecting within the quadtree square

time complexity

- number of subproblems to pursue at each quadtree node $\leq \sum_{i=0}^{8r} \binom{2(4m+4)}{i} i! = (\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})}$
- subproblem at each quadtree node, ignoring the recursive subproblems, can be solved in $((\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})})^{O(1)}$ time
  - assembling: each subproblem solution of 1^{st} child with each ... with each subproblem solution of 4^{th} child
  - while verifying whether the assembled subtour is according to enter-exit constraints and non-intersecting within the quadtree square
- since there are $O(n \lg n)$ nodes in the quadtree, the time is $n(\frac{1}{\epsilon} \lg n)^{O(\frac{1}{\epsilon})}$

- this includes the time to brute-force enumeration at leaf nodes

patching lemma: patch the path with respect to a line segment

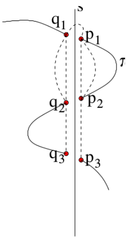
find an Eulerian tour in the graph

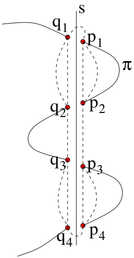
- replace a polyline π that crosses a line segment s at least three times with a polyline π' that crosses s at most twice: construct an Eulerian graph by introducing edges as in the above figure

  $\|\pi'\| \leq \|\pi\| + (\sum_{i \text{ being odd}} \|p_i p_{i+1}\| + \|q_i q_{i+1}\|) + 2\|s\| \leq \|\pi\| + 4\|s\|$

  ———————— (0)

patching lemma: patch the path with respect to a line segment





find an Eulerian tour in the graph

- replace a polyline π that crosses a line segment s at least three times with a polyline π' that crosses s at most twice: construct an Eulerian graph by introducing edges as in the above figure

  $\|\pi'\| \leq \|\pi\| + (\sum_{i \text{ being odd}} \|p_i p_{i+1}\| + \|q_i q_{i+1}\|) + 2\|s\| \leq \|\pi\| + 4\|s\|$ ———————— (0)

- as the length of any edge of G^i is $\frac{1}{2^i}$, error in patching any one edge of G^i is $\leq \frac{4}{2^i}$

patching lemma: patch the path with respect to a line segment

find an Eulerian tour in the graph

- replace a polyline π that crosses a line segment s at least three times with a polyline π' that crosses s at most twice: construct an Eulerian graph by introducing edges as in the above figure

  $\|\pi'\| \leq \|\pi\| + (\sum_{i \text{ being odd}} \|p_i p_{i+1}\| + \|q_i q_{i+1}\|) + 2\|s\| \leq \|\pi\| + 4\|s\|$ ———————— (0)

- as the length of any edge of G^i is $\frac{1}{2^i}$, error in patching any one edge of G^i is $\leq \frac{4}{2^i}$
- viewing the portal is of zero lengthed segment, any optimal solution need to use a portal at most twice (hence, we are allowing two intersections per portal)

patching grid edges at every level of the quadtree

- while considering quadtree nodes from bottom-to-top[3], at every level $H \geq i \geq 1$[4], patch the tour π_{opt} with respect to every grid edge of G^i:

  if the current tour intersects e more than r times so that it intersects e at most twice after patching

[3] when we fix an edge of a grid so that the tour does not intersect it too many times, the number of times the patched tour crosses boundaries of higher-level nodes of the quadtree also goes down; similarly, the total number of crossings (of the tour with the grids) drop exponentially as we use larger and larger grids; thus requiring fewer fix-ups; thus, effectively patching happens in the bottom levels of the quadtree

[4] note G^0 does not intersect the generated path

patching grid edges at every level of the quadtree

- while considering quadtree nodes from bottom-to-top[3], at every level $H \geq i \geq 1$[4], patch the tour π_{opt} with respect to every grid edge of G^i:

  if the current tour intersects e more than r times so that it intersects e at most twice after patching

  let π_i denote the resulting path after patching at G^i

  let π_{i+1} denote the path just before patching at G^i

  π_{H+1} is π_{opt}, and π_1 is the patched tour after patching at every level from H to 1

[3] when we fix an edge of a grid so that the tour does not intersect it too many times, the number of times the patched tour crosses boundaries of higher-level nodes of the quadtree also goes down; similarly, the total number of crossings (of the tour with the grids) drop exponentially as we use larger and larger grids; thus requiring fewer fix-ups; thus, effectively patching happens in the bottom levels of the quadtree

[4] note G^0 does not intersect the generated path

significance of sliding grid

randomly choose the unit grid origin from $[0,0] \times [\frac{1}{2}, \frac{1}{2}]$:

- for every level $i \geq 1$ of the quadtree, the expected number of intersections of a polyline w with the vertical and horizontal edges of G^i is at most $2^{i+1}\|w\|$ ← probabilistic argument is due to sliding grid

  ———— (1)

significance of sliding grid

randomly choose the unit grid origin from $[0,0] \times [\frac{1}{2}, \frac{1}{2}]$:

- for every level $i \geq 1$ of the quadtree, the expected number of intersections of a polyline w with the vertical and horizontal edges of G^i is at most $2^{i+1}\|w\|$ $\leftarrow$ probabilistic argument is due to sliding grid

  ——————— (1)

- if a point p belongs to open edges of G^i then the point p belongs to open edges of G^{i-1} with probability $\frac{1}{2}$ (as every grid edge of G^i has probability $\frac{1}{2}$ to survive and be a grid edge of G^{i-1})

  ——————— (2)

factors causing errors

the length of optimal tour π_{opt} (resp. π_{opt}^{Q}) of P (resp. Q) is denoted by OPT (resp. OPT_Q)

- building a tour π_{opt}^{Q} of points of Q instead of a tour π_{opt} of points of P

factors causing errors

the length of optimal tour π_{opt} (resp. π^{Q}_{opt}) of P (resp. Q) is denoted by OPT (resp. OPT_Q)

- i building a tour π^{Q}_{opt} of points of Q instead of a tour π_{opt} of points of P
- ii limiting the tour to be r-light

factors causing errors

the length of optimal tour π_{opt} (resp. π_{opt}^{Q}) of P (resp. Q) is denoted by OPT (resp. OPT_Q)

- i building a tour π_{opt}^{Q} of points of Q instead of a tour π_{opt} of points of P
- ii limiting the tour to be r-light
- iii limiting tour intersection with the edges of the quadtree squares to portals

factors causing errors

the length of optimal tour π_{opt} (resp. π^Q_{opt}) of P (resp. Q) is denoted by OPT (resp. OPT_Q)

i. building a tour π^Q_{opt} of points of Q instead of a tour π_{opt} of points of P
ii. limiting the tour to be r-light
iii. limiting tour intersection with the edges of the quadtree squares to portals

naturally, each of these is upper bounded by transforming π_{opt} to an approximate tour being computed while forcing it to obey the constraints imposed

type (i) error: rounding points in P to obtain points in Q

- The error in making π_{opt}^{Q} to pass through points of P is

  $\leq n(2\frac{\sqrt{2}}{n\lceil\frac{32}{\epsilon}\rceil})$

  (while walking along π_{opt}^{Q}, for any point $q \in Q$ encountered, walk to q's corresponding location in P and return to q)

  $\leq \frac{4}{32}\epsilon$

  $\leq \frac{\epsilon}{2}OPT$ (since $OPT \geq diam(P)$ which is $\frac{1}{4}$)

type (ii) error: patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

type (ii) error: patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

type (ii) error: patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}]$ (due to (0))

type (ii) error: patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

$$\leq E[2F_1 + \sum_{i=2}^{H} \frac{4F_i}{2^i}]$$

type (ii) error: patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

$$\leq E[2F_1 + \sum_{i=2}^{H} \frac{4F_i}{2^i}]$$

$$\leq E[2F_1] + \frac{4}{r-2} \sum_{i=2}^{H} \frac{E[Y_i]-2E[Y_{i-1}]}{2^i}$$

(noting that due to (2), $E[Y_{i-1}] = E[E[Y_{i-1}|n_i]] = E[\frac{n_i}{2}]$, and

$E[Y_i] = E[\frac{n_i}{2}] \leq E[\frac{Y_i-(r-2)F_i}{2}] = \frac{1}{2}E[Y_i] - \frac{k-2}{2}E[F_i]$)

type (ii) error: patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}]$ (due to (0))

$\leq E[2F_1 + \sum_{i=2}^{H} \frac{4F_i}{2^i}]$

$\leq E[2F_1] + \frac{4}{r-2} \sum_{i=2}^{H} \frac{E[Y_i]-2E[Y_{i-1}]}{2^i}$

(noting that due to (2), $E[Y_{i-1}] = E[E[Y_{i-1}|n_i]] = E[\frac{n_i}{2}]$, and

$E[Y_i] = E[\frac{n_i}{2}] \leq E[\frac{Y_i-(r-2)F_i}{2}] = \frac{1}{2}E[Y_i] - \frac{k-2}{2}E[F_i]$)

$\leq E[2\frac{Y_1}{r-2}] + \frac{4}{r-2}(\frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2})$ (since $F_1 \leq \frac{Y_1}{r} \leq \frac{Y_1}{(r-2)}$)

type (ii) error: patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

$$\leq E[2F_1 + \sum_{i=2}^{H} \frac{4F_i}{2^i}]$$

$$\leq E[2F_1] + \frac{4}{r-2} \sum_{i=2}^{H} \frac{E[Y_i]-2E[Y_{i-1}]}{2^i}$$

(noting that due to (2), $E[Y_{i-1}] = E[E[Y_{i-1}|n_i]] = E[\frac{n_i}{2}]$, and $E[Y_i] = E[\frac{n_i}{2}] \leq E[\frac{Y_i-(r-2)F_i}{2}] = \frac{1}{2}E[Y_i] - \frac{k-2}{2}E[F_i]$)

$$\leq E[2\frac{Y_1}{r-2}] + \frac{4}{r-2}(\frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2}) \quad (\text{since } F_1 \leq \frac{Y_1}{r} \leq \frac{Y_1}{(r-2)})$$

$$= \frac{4}{r-2}(\frac{E[Y_1]}{2} + \frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2})$$

type (ii) error: patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

$$\leq E[2F_1 + \sum_{i=2}^{H} \frac{4F_i}{2^i}]$$

$$\leq E[2F_1] + \frac{4}{r-2} \sum_{i=2}^{H} \frac{E[Y_i]-2E[Y_{i-1}]}{2^i}$$

(noting that due to (2), $E[Y_{i-1}] = E[E[Y_{i-1}|n_i]] = E[\frac{n_i}{2}]$, and
$E[Y_i] = E[\frac{n_i}{2}] \leq E[\frac{Y_i-(r-2)F_i}{2}] = \frac{1}{2}E[Y_i] - \frac{k-2}{2}E[F_i]$)

$$\leq E[2\frac{Y_1}{r-2}] + \frac{4}{r-2}\left(\frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2}\right) \quad \text{(since } F_1 \leq \frac{Y_1}{r} \leq \frac{Y_1}{(r-2)}\text{)}$$

$$= \frac{4}{r-2}\left(\frac{E[Y_1]}{2} + \frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2}\right)$$

$$\leq \frac{4}{r-2}\frac{E[Y_H]}{2^H} \quad \text{(note that } Y_H \text{ is the number of times } \pi_{opt} \text{ crosses the edges of } G^H\text{)}$$

type (ii) error: patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

$$\leq E[2F_1 + \sum_{i=2}^{H} \frac{4F_i}{2^i}]$$

$$\leq E[2F_1] + \frac{4}{r-2} \sum_{i=2}^{H} \frac{E[Y_i] - 2E[Y_{i-1}]}{2^i}$$

(noting that due to (2), $E[Y_{i-1}] = E[E[Y_{i-1}|n_i]] = E[\frac{n_i}{2}]$, and $E[Y_i] = E[\frac{n_i}{2}] \leq E[\frac{Y_i - (r-2)F_i}{2}] = \frac{1}{2}E[Y_i] - \frac{k-2}{2}E[F_i]$)

$$\leq E[2\frac{Y_1}{r-2}] + \frac{4}{r-2}(\frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2}) \quad \text{(since } F_1 \leq \frac{Y_1}{r} \leq \frac{Y_1}{(r-2)}\text{)}$$

$$= \frac{4}{r-2}(\frac{E[Y_1]}{2} + \frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2})$$

$$\leq \frac{4}{r-2}\frac{E[Y_H]}{2^H} \quad \text{(note that } Y_H \text{ is the number of times } \pi_{opt} \text{ crosses the edges of } G^H\text{)}$$

$$= \frac{4}{r-2}\frac{2^{H+1}OPT}{2^H} \quad \text{(due to (1))}$$

type (ii) error: patching π_{opt} at every level from bottom-to-top

F_i: number of patching operations performed at i^{th} level; Y_i: number of times the current tour crosses the edges of G^i just before considering G^i for patching; n_i: number of times the current tour crosses the edges of G^i after patching

expected increase in length of π_{opt} due to patchings at every level

$$= E[\sum_{i=1}^{H} \frac{4F_i}{2^i}] \quad \text{(due to (0))}$$

$$\leq E[2F_1 + \sum_{i=2}^{H} \frac{4F_i}{2^i}]$$

$$\leq E[2F_1] + \frac{4}{r-2} \sum_{i=2}^{H} \frac{E[Y_i]-2E[Y_{i-1}]}{2^i}$$

(noting that due to (2), $E[Y_{i-1}] = E[E[Y_{i-1}|n_i]] = E[\frac{n_i}{2}]$, and $E[Y_i] = E[\frac{n_i}{2}] \leq E[\frac{Y_i-(r-2)F_i}{2}] = \frac{1}{2}E[Y_i] - \frac{k-2}{2}E[F_i]$)

$$\leq E[2\frac{Y_1}{r-2}] + \frac{4}{r-2}(\frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2}) \quad \text{(since } F_1 \leq \frac{Y_1}{r} \leq \frac{Y_1}{(r-2)}\text{)}$$

$$= \frac{4}{r-2}(\frac{E[Y_1]}{2} + \frac{E[Y_H]}{2^H} - \frac{E[Y_1]}{2})$$

$$\leq \frac{4}{r-2}\frac{E[Y_H]}{2^H} \quad \text{(note that } Y_H \text{ is the number of times } \pi_{opt} \text{ crosses the edges of } G^H\text{)}$$

$$= \frac{4}{r-2}\frac{2^{H+1}OPT}{2^H} \quad \text{(due to (1))}$$

$$= \frac{8}{r-2}OPT$$

type (iii) error: expected error due to snapping tour to portals

- as noted, after applying patching lemma to grid squares at all levels from bottom-to-top, the tour intersects with any edge of any grid square at most twice

[5] for $i \geq 1$, tour π_i (esp., π_1) has at least as many crossings with G_i as π_{opt} has with G_i

[6] if a grid edge has portals from several levels, the snapped tour used only the portals on this edge that belong to the highest level

type (iii) error: expected error due to snapping tour to portals

- as noted, after applying patching lemma to grid squares at all levels from bottom-to-top, the tour intersects with any edge of any grid square at most twice

- as the distance between any two successive portals along any edge of G^i is $\frac{1/2^i}{m+1}$, a single snapping operation on an edge of G^i introduces an error $\leq 2(\frac{1/2^i}{2(m+1)}) = \frac{1}{2^i(m+1)}$

[5] for $i \geq 1$, tour π_i (esp., π_1) has at least as many crossings with G_i as π_{opt} has with G_i

[6] if a grid edge has portals from several levels, the snapped tour used only the portals on this edge that belong to the highest level

type (iii) error: expected error due to snapping tour to portals

- as noted, after applying patching lemma to grid squares at all levels from bottom-to-top, the tour intersects with any edge of any grid square at most twice

- as the distance between any two successive portals along any edge of G^i is $\frac{1/2^i}{m+1}$, a single snapping operation on an edge of G^i introduces an error $\leq 2(\frac{1/2^i}{2(m+1)}) = \frac{1}{2^i(m+1)}$

- let Z_i be the number of intersections of π_{opt} with G^i for $i \geq 1$ [5]

  due to (1), $E[Z_i] \leq 2^{i+1} OPT$

[5] for $i \geq 1$, tour π_i (esp., π_1) has at least as many crossings with G_i as π_{opt} has with G_i

[6] if a grid edge has portals from several levels, the snapped tour used only the portals on this edge that belong to the highest level

type (iii) error: expected error due to snapping tour to portals

- as noted, after applying patching lemma to grid squares at all levels from bottom-to-top, the tour intersects with any edge of any grid square at most twice

- as the distance between any two successive portals along any edge of G^i is $\frac{1/2^i}{m+1}$, a single snapping operation on an edge of G^i introduces an error $\leq 2(\frac{1/2^i}{2(m+1)}) = \frac{1}{2^i(m+1)}$

- let Z_i be the number of intersections of π_{opt} with G^i for $i \geq 1$ [5]

  due to (1), $E[Z_i] \leq 2^{i+1} OPT$

- expected increase in length due to snapping π_1 to the portals of the quadtree (summed over all the levels)[6] $\leq E[\sum_{i=1}^{H} \frac{Z_i}{2^i(m+1)}] \leq \frac{2H}{m+1} OPT$

[5] for $i \geq 1$, tour π_i (esp., π_1) has at least as many crossings with G_i as π_{opt} has with G_i

[6] if a grid edge has portals from several levels, the snapped tour used only the portals on this edge that belong to the highest level

overall approximation factor

- $(1+\frac{\epsilon}{2})(1+\frac{8}{r-2})(1+\frac{2H}{m+1})OPT$

  $\leq (1+\frac{\epsilon}{2})(1+\frac{\epsilon}{10})^2 OPT$ (by choosing $r = \frac{90}{\epsilon} = O(\frac{1}{\epsilon})$ and $m \geq \frac{20H}{\epsilon} = O(\frac{\lg n}{\epsilon})$)

  $\leq (1+\epsilon)OPT$

overall approximation factor

- $(1+\frac{\epsilon}{2})(1+\frac{8}{r-2})(1+\frac{2H}{m+1})OPT$

  $\leq (1+\frac{\epsilon}{2})(1+\frac{\epsilon}{10})^2 OPT$ (by choosing $r=\frac{90}{\epsilon}=O(\frac{1}{\epsilon})$ and $m \geq \frac{20H}{\epsilon}=O(\frac{\lg n}{\epsilon})$)

  $\leq (1+\epsilon)OPT$

hence, the achieved approximation factor in expectation is $(1+\epsilon)$, and the time complexity is $n(\frac{\lg n}{\epsilon})^{O(1/\epsilon)}$

take-home lessons

- rounding input points
- Steiner points (portals) along grid edges
- dynamic programming
- locally patching a path
- grid shifting
- technique is applicable to compute various other structures:
  - for example, minimum Steiner tree, k-TSP, k-MST, minimum cost perfect matching

hence, the Godel award for the result!

references

- S. Har-Peled. Geometric Approximation Algorithms. American Mathematical Society, 2011.
- D. P. Williamson and D. B. Shmoys. The Design of Approximation Algorithms. Cambridge University Press, 2011.
- S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. Journal of ACM, 1998.

references

- S. Har-Peled. Geometric Approximation Algorithms. American Mathematical Society, 2011.
- D. P. Williamson and D. B. Shmoys. The Design of Approximation Algorithms. Cambridge University Press, 2011.
- S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. Journal of ACM, 1998.

Parallel/further significant work -

- J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple new method for the geometric TSP, k-MST, and related problems. SIAM Journal on Computing, 1999. $\leftarrow O(n^{d+1} polylog)$ time PTAS
- S. B. Rao and W. D. Smith. Approximating geometric graphs via ”spanners” and ”banyans”. Symposium on Theory of Computing, 1998. $\leftarrow O(n \lg n)$ time PTAS
- L. Trevisan. When Hamming meets Euclid: the approximability of geometric TSP and Steiner tree. SIAM Journal on Computing, 2000. $\leftarrow$ NP-hard to get $(1+\epsilon)$-approx in $\mathbb{R}^{O(\lg n)}$, for some $\epsilon > 0$