On Some Geometric Optimization Problems:
Some Recent Results and Continuing Challenges

Joe Mitchell

State University of New York
Stony Brook, NY 11794–3600
Example Problem 1:

Shortest Paths Among Obstacles: Visibility Graph

Air Traffic Management: Multiple “well separated” routes among moving obstacles
Example Problem 2:

Visit a set (unordered) or a sequence (ordered) of cities/regions
Example Problem 3:

Optimally mow a lawn or mill a pocket

Worry about removing material: “Snow blower problem”
Example Problem 4:

Search a building/view or map a region: Watchman Route
Example Problem 5:

Awaken a swarm of robots: “Freeze-Tag” Problem
General Geometric Optimal Tour Problem:

Given a geometric domain $\mathcal{D} \subset \mathbb{R}^d$, find an “optimal” tour/path in $\mathcal{D}$ that satisfies a given set of constraints:

- visit each element of a set $S$ of points, lines, regions, etc.
- “see” all of $\mathcal{D}$
- mill/mow/sweep the domain $\mathcal{D}$
- pass through a given root point
“Optimal” can have various meanings:

- shortest total Euclidean length
- shortest total $L_p$ length
- minimum number of links (link metric)
- minimum amount of turning
- min the max edge length
- max the min edge length
- min/max area of the enclosed region
- etc.
Approximation Algorithms:

\textit{c-approximation}: cost at most \( c \) times optimal, for a minimization problem \((c > 1)\)

\textit{Polynomial Time Approximation Scheme} (PTAS):
method giving \((1 + \epsilon)\)-approx to the optimal (minimum), in time polynomial in \( n \), for \textit{any} fixed \( \epsilon > 0 \).

Dependence on \( \epsilon \) may be exponential in \((1/\epsilon)\); else \textit{FPTAS}
Overview of Talk:

- Approximation algorithms: Classical TSP
- Visiting a sequence of regions: Touring Polygons Problem (TPP)
- 3D shortest path problems
  - Some “simple” 3D shortest path problems that are hard
  - Poly-time solutions to some other “simple” 3D path problems
- A TSP variant: TSP with Neighborhoods
- Min-diameter bounded degree spanning trees: Freeze-Tag Problem
Classical TSP on Points:

- $S = \text{set of } n \text{ points in } \mathbb{R}^d$
- NP-hard
- $n^{O(n^{1-1/d})}$ exact (subexponential) \[ \text{[SmWo98]} \]
- Simple 2-approx: double the MST and shortcut (holds in metric spaces)

- Christofides: 1.5-approx
  (use MST $\cup$ min-weight matching on odd-degree nodes of MST)
A Recipe for Approximation Algorithms:

- Show how to convert $\text{OPT}$ to a special class $C$ of solution instances
  
  Give THM showing that this conversion does not cost much
  
  $L_{\text{OPT}}^C \leq c_1 \cdot L^*$

- Give an efficient algorithm for exact optimization over class $C$

- Show how to recover a feasible solution to the original problem at small cost
  
  $L_A \leq c_2 \cdot L_C^*$
A Recipe for Approximation Algorithms (cont):

Overall, then the approx solution has cost

\[ L_A \leq c_2 \cdot L_C^* \leq c_2 \cdot L_C^{OPT} \leq c_2 \cdot c_1 \cdot L^* \]
PTAS Results for Classical TSP:

- $O(n^{O(1/\epsilon)})$ in $\mathbb{R}^2$  \hspace{1cm} [Ar96, Mi96]
- $O(n^{O(1)})$ in $\mathbb{R}^2$ \hspace{1cm} [Mi97]
- $O(n(\log n)^{(O(d))^{d-1}})$ expected \hspace{0.5cm} $O(n^{d+1}\text{polylog})$ det.) \hspace{1cm} [Ar97]
- $O(n \log n)$ deterministic \hspace{1cm} [RaSm98]
  
  Idea: $t$-spanners and “$t$-banyons”

- NP-hard to get $(1 + \epsilon)$-approx in $\mathbb{R}^{O(\log n)}$, for some $\epsilon > 0$ \hspace{1cm} [Tr97]

- MAX-SNP-hard in metric spaces
  
  No $c$-approx for $c < 129/128$ \hspace{0.5cm} ($c < 41/40$, asym.) \hspace{1cm} [PV99]
Main Idea of PTAS’s:

Transform $\text{OPT}$ into a near-opt network of special recursive structure that allows efficient optimization by dynamic programming
Simple PTAS in $\mathbb{R}^2$:

Special class $C$: “$m$-guillotine subdivisions”

**Theorem:** For any polygonal subdivision $S$ (edges $E$) of length $L$, and any integer $m > 0$, there exists an $m$-guillotine subdivision, $S_G$ (edges $E_G$) of length at most $(1 + \frac{\sqrt{2}}{m})L$, with $E \subseteq E_G$.

[Rectilinear subdivisions: factor $(1 + \frac{1}{m})$]

**Algorithm:** Optimizing over the class of $m$-guillotine subdivisions is easily done using dynamic programming (DP) algorithms.

Algorithm produces $S^*_G$, a *shortest* $m$-guillotine subdivision that obeys certain properties; e.g., connected, spanning all or some ($\geq k$) of the input points

**Solution Recovery** from $S^*_G$:

- constraint in DP forcing Eulerian spanning subgraph of $S^*_G$
Example: 3-Guillotine Subdivision:
Definitions: $m$-span:

\[ W \cap \ell \cap E \quad (\xi = 6) \]

\[ \sigma_1(\ell) \quad \sigma_2(\ell) \quad \sigma_3(\ell) \quad \sigma_4(\ell) = \sigma_5(\ell) = \cdots = \emptyset \]
Definitions: \( m \)-perfect:

Example:
\( \ell_1, \ell_2, \ell_3, \ell_4 \) are 3-perfect (also \( m \)-perfect, \( m \geq 4 \))
\( \ell_4 \) is also 2-perfect (but not 1-perfect)
Definitions: \textit{m-Guillotine:}

Subdivision $S$ is \textit{m-guillotine} wrt $W$ if either

(1) $V \cap \text{int}(W) = \emptyset$

\textit{OR}

(2) $\exists$ \textit{m-perfect cut} $\ell$ (wrt a minimal window, $\overline{W} \subseteq W$) such that $S$ is \textit{m-guillotine} wrt $W \cap H^+$ and $W \cap H^-$

($H^+$, $H^-$ are the closed halfplanes induced by $\ell$)
Proof of Main Theorem:

Goal: Add new edges of total length $\leq \left(\frac{1}{m}\right)L$ to make $S$ $m$-guillotine

Recursive construction: At each stage, $\exists$ a cut, $\ell$, such that we can add (if necessary) the $m$-span of $\ell$ to $E$, making $\ell$ $m$-perfect.

Charging scheme: For each unit of length we add (when we add $\sigma_m(\ell)$), there are $m$ units of length of $E$ to “pay” for it.

If $m$-perfect cut exists, use it, and recurse.
Thus, assume no $m$-perfect cut exists.
Proof of Main Theorem:

Consider making a horizontal cut, $\ell$.

- **What does it “cost”?**

  \textit{Ans:} We \textit{add} (at most) the length of the $m$-span, $\sigma_m(\ell)$

- **Who can pay for it?**

  \textit{Ans:} The “$m$-dark” points along $\ell$
If horiz subseg $\overline{pq}$ (of length $\lambda$) is $m$-dark, we charge $\lambda$ off to the bottoms of the first $m$ subsegs above $\overline{pq}$, and the tops of the first $m$ subsegs below $\overline{pq}$. Each level above/below “pays” $\frac{\lambda}{2m}$.

**Example:** Portion $\overline{pq} \subset \ell$ is 3-dark
Key Lemma: For any subdivision $S$ and any window $W$, there exists a horiz/vert “favorable” cut whose $m$-dark portion is at least as long as its $m$-span.

For a favorable cut $\ell$, we add its span to the edge set, and recurse on each side of $\ell$.

Note: After a portion of $E$ has been charged on one side, it will not be charged again on that side, since it will be within $m$ levels of the boundary of any future windows.

Thus, no portion of $E$ charged more than its length times $\frac{1}{2m} + \frac{1}{2m}$

Thus, the total length added to $E$ is at most $(\frac{1}{m})L$. 
Proof of Key Lemma:

**Key Lemma:** For any subdivision $S$ and any window $W$, there exists a horiz/vert “favorable” cut whose $m$-dark portion is at least as long as its $m$-span.

- Let $f(x) =$ length of $m$-span of vertical line through $x$
  Let $g(y) =$ length of $m$-span of horizontal line through $y$

- Then,
  $$A_x = \int f(x) \, dx$$
  is simply the area of the “$m$-dark” (RED) region wrt horiz cuts
  Similarly,
  $$A_y = \int g(y) \, dy$$
  is the area of the “$m$-dark” (BLUE) region wrt vertical cuts

- Assume, WLOG, that $A_x \geq A_y$
Thus, for $h(y) = \text{length of } m\text{-dark}, \text{ for horiz line through } y,$

\[ A_x = \int h(y)dy \geq \int g(y)dy = A_y > 0 \]

So, $\exists y^* \text{ for which } h(y^*) \geq g(y^*);$  
i.e., $\exists \text{ a horiz line through } y^* \text{ whose } m\text{-dark portion } \geq m\text{-span}.$

(If $A_x \leq A_y$, then $\exists \text{ a vertical favorable cut}.$)
A rectilinear subdivision.
Region of 2-dark points wrt horizontal cuts (RED)

\[ f(x) \]
Region of 2-dark points wrt vertical cuts (BLUE)
Example Application: Steiner Tree:

Given: set $P$ of $n$ points (WLOG: no two have common $x$- or $y$-coord)

Consider the rectilinear ($L_1$) case first.

Let $T_R = a$ min-length rectilinear Steiner tree for $P$; length $L^*$

By the Main Theorem, $\exists m$-guillotine subdiv $S_G$, with edge set $E_G$, of length at most $(1 + \frac{1}{m})L^*$, such that $T_R \subseteq E_G$.

$\exists$ recursive cutting of $S_G$ into windows, each side with $\leq 2m$ endpts

Thus, we can partition into subproblems having small descriptions.
Dynamic Programming Algorithm (Steiner):

Sorted $x$-coord: $x_1 < x_2 < \cdots$ (for $P$, and grid lines)
Sorted $y$-coord: $y_1 < y_2 < \cdots$

**Subproblem:** $O(n^4 \cdot (n^{2m})^4) = O(n^{8m+4})$ choices

**Input:**

1. a rectangle $R(i, i', j, j')$, defined by $x_i, x_{i'}, y_j, y_{j'}$ \(O(n^4)\)

2. four sets of “boundary information”, $\Sigma_l$, $\Sigma_r$, $\Sigma_b$, and $\Sigma_t$, determined by $\leq 2m$ endpoints on each side \(O((n^{2m})^4)\)

3. a partition, $\mathcal{P}$, of $\cup_{\alpha} \Sigma_{\alpha}$, giving required connectivity among boundary pieces \(O(1)\)

**Objective:** Find min-length $m$-guillotine subdivision, $S^*_G$ (edges $E^*_G$ interior to $R(i, i', j, j')$), such that $E^*_G$ covers $P$ and $E^*_G$ connects the boundary pieces, according to partition $\mathcal{P}$.
\[ R(i, i', j, j') \]
Dynamic Programming Algorithm (Steiner):

Let $V =$ opt value of a subproblem

*IF* all connections required by $\mathcal{P}$ already satisfied by boundary segments

*THEN*

$V = 0$

*ELSE*

Compute $V$ recursively as the sum of the values of two subproblems obtained by splitting, minimized over:

1. $O(n)$ choices of a hor/vert cut

2. $O(n^2m)$ choices of new boundary segments, $\Sigma$, on the cut

3. $O(1)$ choices of (compatible) partitions for resulting boundary segments in two new subproblems

Running time: — $O(n^{10m+5})$
**Corollary:** For any fixed positive integer $m$, there is an $O(n^{10m+5})$ algorithm to compute an approx Steiner tree whose $L_1$ length is within a factor $(1 + \frac{1}{m})$ of optimal.

Euclidean case: Same idea, but add a regular grid of potential (approx) Steiner points for $P$ (spacing $\frac{\text{diam}(P)}{nM}$, for some $M \geq m$).

**Corollary:** For any fixed positive integer $m$, there is an $O(n^{O(m)})$ algorithm to compute a Steiner spanning tree (or Steiner $k$-MST) whose length is within a factor $(1 + \frac{1}{m})$ of minimum.
Minimum-Weight Triangulation:

Given $n$ points in the plane, compute a convex subdivision or a triangulation of minimum total edge length.

Two cases:  \textbf{With or Without Steiner points}

\textbf{OPEN:} Complexity of Minimum-Weight Triangulation (MWT)?
Minimum-Weight Triangulation:

Given \( n \) points in the plane, compute a convex subdivision or a triangulation of minimum total edge length.

Two cases: **With or Without Steiner points**

**OPEN:** Complexity of Minimum-Weight Triangulation (MWT)?

*Brand new result!* Proved NP-hard by Mulzer and Rote (2006)

**OPEN:** Does a min-weight Steiner triangulation exist?

- MWT has \( O(1) \)-approx
  
  **OPEN:** PTAS for MWT?

- Steiner MWT has \( O(1) \)-approx
  
  **OPEN:** PTAS for Steiner MWT?
• Min-weight Steiner convex subdivision has PTAS

**OPEN:** PTAS for min-weight convex subdivision (no Steiner points)?
The Touring Polygons Problem (TPP)  

[Dror-Efrat-Lubiw-M]:

Given a sequence of $k$ polygons in the plane, a start point $s$, and a target point, $t$, we seek a shortest path that starts at $s$, visits in order each of the polygons, and ends at $t$. 

![Diagram of polygons and start and target points]
Related Problem: TSPN:

If the order to visit \( \{P_1, P_2, \ldots, P_k\} \) is not specified, we get the NP-hard TSP with Neighborhoods problem.

TSPN: \( O(\log n) \)-approx in general

\( O(1) \)-approx, PTAS in special cases
The Fenced Problem:

Here that part of the path connecting $P_i$ to $P_{i+1}$ must lie inside a simple polygon $F_i$, called the fence.
The Fenced Problem:
The Fenced Problem:
The Fenced Problem:
The Fenced Problem:
The Fenced Problem:
Applications: Parts Cutting Problem:
Applications: Safari Problem:
Applications: Zookeeper Problem:
Fact: The optimal path visits the essential cuts in the order they appear along $\partial P$. 
Summary of TPP Results:

- Disjoint convex polygons: $O(kn \log(n/k))$ time, $O(n)$ space
  (For fixed $s$, $\{P_1, P_2, \ldots, P_k\}$, $O(k \log(n/k))$ shortest path queries to $t$.)
- Arbitrary convex polygons: $O(nk^2 \log n)$ time, $O(nk)$ space
- Full combinatorial map: worst-case size $\Theta((n - k)2^k)$
  Output-sensitive algorithm; $O(k + \log n)$-time shortest path queries.
- TPP for nonconvex polygons: NP-hard
  FPTAS, as special case of 3D shortest paths
Applications:

- Safari: $O(n^2 \log n)$ vs. $O(n^3)$
- Watchman: $O(n^3 \log n)$ vs. $O(n^4)$
  
  floating watchman: $O(n^4 \log n)$ vs. $O(n^5)$

  We avoid use of complicated path “adjustments” arguments, DP

- Parts cutting: $O(kn \log(n/k))$
Relationship to 3D Shortest Paths:
Relationship to 3D Shortest Paths:
Relationship to 3D Shortest Paths:

We show:

- Holes are convex: poly-time
- Non-convex holes: NP-hard
Unconstrained TPP: Disjoint Convex Polygons:

Given: \( s, t \), sequence of disjoint convex polygons \( (P_1, \ldots, P_k) \)
Goal: Find a shortest \( k \)-path from \( s = P_0 \) to \( t \).
Local Optimality Conditions:
Unconstrained TPP: Disjoint Convex Polygons:

**Lemma:** For any $t \in \mathbb{R}^2$ and any $i \in \{0, \ldots, k\}$, $\exists$ unique shortest $i$-path, $\pi_i(p)$, from $s = P_0$ to $t$.
Thus, local optimality is equivalent to global optimality.

**Lemma:** In the TPP for disjoint convex polygons ($P_1, \ldots, P_k$), each first contact set $T_i$ is a (connected) chain on $\partial P_i$.

**Lemma:** For any $p \in \mathbb{R}^2$ and any $i$, there is a unique point $p' \in T_i$ such that $\pi_i(p) = \pi_{i-1}(p') \cup p'p$. 
General Approach: Build a Shortest Path Map:

SPM\(_k\)(s): a decomposition of the plane into cells according to the combinatorial type of a shortest \(k\)-path to \(t\)

**Bad news**: worst-case size can be huge:

**Theorem**: The worst-case complexity of SPM\(_k\)(s) is \(\Omega((n - k)2^k)\)
$s \cdot 2^i \rightarrow 2^{i+1}$
Good news: worst-case size cannot be \textit{bigger} than “huge”:

\textbf{Theorem:} The worst-case complexity of $\text{SPM}_k(s)$ is $O((n - k)2^k)$

Size $m_i$ satisfies $m_i \leq 2m_{i-1} + O(|P_i|)$.

Output-sensitive algorithm to build SPM:

\textbf{Theorem:} One can compute $\text{SPM}_k(s)$ in time $O(k \cdot |\text{SPM}_k(s)|)$, after which a shortest $k$-path from $s$ to a query point $t$ can be computed in time $O(k + \log n)$. 
Last Step Shortest Path Map:

\( T_i = \text{first contact set of } P_i: \) points where a shortest \((i - 1)\)-path first enters \( P_i \) after visiting \( P_1, \ldots, P_{i-1} \)

For \( p \in T_i:\)
- \( r^s_i(p) = \text{set of rays of locally shortest } i\text{-paths going straight through } p: \) a single ray
- \( r^b_i(p) = \text{set of rays of locally shortest } i\text{-paths properly reflecting at } p \) a single ray (\( p \) interior to an edge of \( T_i \)), or a cone (\( p \) a vertex of \( T_i \))
- \( r_i(p) = r^s_i(p) \cup r^b_i(p) \)

\( R_i = \bigcup_{p \in T_i} r_i(p) \) (an infinite family of rays) is the starburst with source \( T_i \)
The Last Step Shortest Path Map:

\( S_i = \) the last step shortest path map, subdivision according to the
combinatorial type of the rays of \( R_i \) passing through points \( p \in \mathbb{R}^2 \)

\( S_i \) decomposes the plane into cells \( \sigma \) of two types:

1. cones with an apex at a vertex \( v \) of \( T_i \), whose bounding rays are
   reflection rays \( r'_1(v) \) and \( r'_2(v) \)
   \( v \) is the source of cell \( \sigma \)

2. unbounded 3-sided regions associated with edge \( e \) of \( T_i \), classified as
   - reflection cells or
   - pass-through cells

\( e \) is the source of cell \( \sigma \)

The pass-through region is the union of all pass-through cells
Last Step Shortest Path Map:

Pass-through Region
Using the Last Step Shortest Path Map:

Find a shortest $i$-path to query point $q$:

Locate $q$ in $S_i$

- cell $\sigma$ rooted at vertex $v$ of $T_i$
  - last segment of $\pi_i(q)$ is $\overline{vq}$
  - recursively compute $\pi_{i-1}(v)$ (locate $v$ in $S_{i-1}$, etc)

- cell $\sigma$ rooted at edge $e$ of $T_i$
  - $\sigma$ is pass-through: $\pi_i(q) = \pi_{i-1}(q)$, so recursively compute shortest $(i-1)$-path to $q$
  - $\sigma$ is a reflection cell: recursively compute shortest $(i-1)$-path to $q'$, the reflection of $q$ wrt $e$

Lemma: Given $S_1, \ldots, S_i$, $\pi_i(q)$ can be determined in time $O(k \log(n/k))$
Algorithm:

Construct each of the subdivisions $S_1, S_2, \ldots, S_k$ iteratively:

For each vertex $v_j$ of $P_{i+1}$, we compute $\pi_i(v_j)$.

If this path arrives at $v_j$ from the inside of $P_{i+1}$, then $v_j$ is not a vertex of $T_{i+1}$.

Otherwise it is, and the last segment of $\pi_i(v_j)$ determines the rays $r_{i}^{b}(v_j)$ and $r_{i}^{s}(v_j)$ that define the subdivision $S_{i+1}$.

**Theorem:** For a given sequence $(P_1, \ldots, P_k)$ of $k$ disjoint convex polygons having a total of $n$ vertices, a data structure of size $O(n)$ can be constructed in time $O(kn \log(n/k))$ that enables shortest $i$-path queries to any query point $q$ to be answered in time $O(i \log(n/k))$. 
TPP for Fenced, Arbitrary Convex Polygons:

Use Last Step Shortest Path Maps, but combinatorics and algorithm are substantially more complex.

Needed for Safari, Watchman Route, Zookeeper.
Proposition: The TPP in the $L_1$ metric is polynomially solvable (in $O(n^2)$ time and space) for arbitrary rectilinear polygons $P_i$ and arbitrary fences $F_i$. The result lifts to any fixed dimension $d$ if the regions $P_i$ and the constraining regions $F_i$ are orthohedral.
TPP on Nonconvex Polygons:

Theorem: The touring polygons problem is NP-hard, for any $L_p$ metric ($p \geq 1$), in the case of nonconvex polygons $P_i$, even in the unconstrained ($F_i = \mathbb{R}^2$) case with obstacles bounded by edges having angles 0, 45, or 90 degrees with respect to the $x$-axis.

Proof: from 3-SAT
based on a careful adaptation of Canny-Reif proof
Open Problems:

OPEN: What is the complexity of the TPP for disjoint non-convex simple polygons?

OPEN: Complexity of TPP if path must be simple?

OPEN: Compute a simple path of min length linking points $p_1, p_2, \ldots, p_n$ in order?

OPEN: $O(1)$-approx for watchman routes in polygons with holes?

OPEN: Any watchman route approx in 3D?

OPEN: Multiple watchmen?
3D Shortest Paths: Background:

- NP-hard in general
- FPTAS
  
  \((1 + \epsilon)\text{-approx in time poly}(n, 1/\epsilon)\)

- Special cases: surfaces, \(k\) convex polytopes, buildings of \(k\) heights
  
  \((\text{time } O(n^{O(k)}))\)
Shortest Paths Among Stacked (Flat) Obstacles:

If obstacles are \textit{complements} of convex polygons: TPP solves (case includes halfplanes)
Shortest Paths Among Stacked (Flat) Obstacles:

If obstacles are *complements* of convex polygons: TPP solves 
(*case includes halfplanes*)

What if obstacles are convex polygons?
Shortest Paths Among Stacked (Flat) Obstacles:

If obstacles are *complements* of convex polygons: TPP solves
(case includes halfplanes)

What if obstacles are convex polygons?
Canny-Reif: NP-hard for stacked 45-45-90 triangles
Shortest Paths Among Stacked (Flat) Obstacles:

If obstacles are *complements* of convex polygons: TPP solves (case includes halfplanes)

What if obstacles are convex polygons?
Canny-Reif: NP-hard for stacked 45-45-90 triangles
What about axis-aligned rectangular obstacles?
Shortest Paths Among Stacked (Flat) Obstacles:

If obstacles are **complements** of convex polygons: TPP solves (case includes halfplanes)

What if obstacles are convex polygons?
Canny-Reif: NP-hard for stacked 45-45-90 triangles

What about axis-aligned rectangular obstacles?
New result: Still NP-hard

[M-Sharir]
Hardness Proof:

Theorem: The Euclidean shortest path problem is NP-hard for a stack of axis-parallel rectangles as obstacles.

Proof: from 3-SAT, based on modified Canny-Reif proof

- Use a cascade of path splitter gadgets to get $2^n$ combinatorially distinct path classes
  Paths encode an assignment of the $n$ variables: path # $i$ encodes assignment given by the $(n$-bit) binary representation of $i$.

- Use path shuffle gadgets to rearrange paths within a class

- Use shuffle gadgets to construct a literal filter: the only path classes that pass through unobstructed are those having bit $b_i$ set accordingly

- Assemble 3 literal filters per clause filter: output of clause filter will contain short path classes only for those assignments (if any) that satisfy the instance of 3SAT

- Collect paths back into one path class, using inverted path splitting gadgets.
Final question: Is there a path from \( s \) to \( t \) of length \( L \)?
Yes, iff the formula is satisfied.
Path Splitting Gadget:
Path Splitting Gadget:
Path Splitting Gadget:
Instances of Stacked Obstacles:

Poly-Time

NP-Complete

Poly time when the obstacles are “terrain like”
(e.g., all contain a downwards ray)
Shortest Paths Among Balls:

Also NP-Hard: $L_1$ shortest paths among balls in 3D

**OPEN:** Euclidean shortest paths among balls in 3D? Unit balls?

**OPEN:** Euclidean shortest paths among aligned cubes in 3D? Unit cubes?
Shortest Path Over Walls

[M-Sharir]:

$n$ lines in 3D: $e_1, \ldots, e_n$, each $\perp$ to $y$-axis

$e_i: y = a_i, z = b_ix + c_i$, with $a_1 < a_2 < \cdots < a_n$

Each $e_i$ defines a (vertical halfplane) wall $W_i$

Goal: Find $L_2$-shortest path from $s$ to $t$ avoiding the interiors of walls
Some properties of $\pi(\cdot)$ and $L(\cdot)$:

$\pi(\zeta) =$ shortest path from $s$ to $\zeta \in \mathbb{R}^3$
$L(\zeta) =$ length of path $\pi(\zeta)$

(1) $\pi(\zeta)$ is $y$-monotone, polygonal, bending on some of the edges $e_i$

(2) $\pi(\zeta) = \pi_1 \parallel \pi_2$, with $\pi_1$ ascending (in $z$), $\pi_2$ descending

(3) The path $\pi(\zeta)$ is unique
   Corollary: As $\zeta$ varies along a line $\ell$,
   (i) $L(\zeta)$ is a convex function of $\zeta \in \ell$
   (ii) $\pi(\zeta)$ varies continuously (Hausdorff metric)
   (iii) The combinatorial structure of $\pi(\zeta)$ changes only when it passes through
       3 collinear (mutually visible) points on 3 distinct edges

(4) Solution by convex programming: LP-type problem
The Shortest Path Map:

Combinatorial Complexity of the Shortest-Path Map:

**Lemma:** For each $i < n$, the set

$$C_i = \{ \zeta \in e_n \mid \pi(\zeta) \cap e_i \neq \emptyset \}$$

is connected.

**Theorem:** The number of combinatorial changes in the structure of $\pi(\zeta)$, as $\zeta$ moves along $e_n$, is $O(n)$. 
**$L_1$ Shortest Paths Over Terrains**

Structure: Can assume path goes up from $s$ to $s'$ (altitude $h$), then along a shortest path in the plane $z = h$ to $t'$, then down to $t$

Atomic intervals: Partition heights $h$ according to vertex heights ($v_z$), and critical heights at which $\exists$ edges $e, e'$ of $T$ for which the points $e(h)$ and $e'(h)$ have the same $x$- or $y$-coordinate.

$O(n^2)$ atomic intervals
$L_1$ Shortest Paths Over Terrains (cont):

**Lemma:** The length function, $L(h)$, is concave, piecewise-linear over each atomic interval.

**Algorithm:** Compute shortest path for each of the $O(n^2)$ critical heights, and take best: $O(n^3 \log n)$.

**Work in progress:** Analyzing the SPM for $L_1$ shortest paths over a terrain.

**OPEN:** Euclidean shortest paths over a terrain?

Note: SPM has worst-case exponential size.
TSP with Neighborhoods (TSPN):

\[ S = \{X_1, X_2, \ldots, X_k\} \], a set of regions that must be visited
TSP with Neighborhoods (cont):
TSP with Neighborhoods (cont):

Problem introduced by Arkin-Hassin

- “obvious” heuristics do not work:
  - TSP approx on centroids (as representative points)
  - greedy algorithms (Prim- or Kruskal-like)

- $O(1)$-approx for “nice” regions:
  - (a) parallel unit segments
  - (b) unit disks
  - (c) translates of a polygon $P$

- “Combination Lemma”
TSPN – More Approximation Results:

• General (connected) regions: $O(\log k)$-approx
  use guillotine rectangular subdivisions (GRS), DP
  $O(n^5)$ time
  (difficulty is in DP)

• Improved running time, using modified GRS
  $O(n + k \log k)$
Difficulty in Applying TSP Methods to TSPN:

How to define a subproblem succinctly?

Which regions must be visited inside R?
Recent Progress on TSPN:

- $O(1)$-approx for regions of comparable size (diameter) [DM]
- PTAS for disjoint fat regions of comparable size [DM]
  (use a new charging scheme and $m$-guillotine subdivisions)
  Generalized to point clusters, $\mathbb{R}^d$ [FeGr]
- $O(1)$-approx for disjoint fat regions of any size [BGK+]
- PTAS for disjoint fat regions of any size [Mi06]
  (use new structural results and a new charging scheme)
Recent Progress on TSPN (cont.):

- Hardness of approx:
  
  No $c$-approx with $c < 391/390$, unless $P=NP$  
  
  No $c$-approx with $c < 2$, unless $NP \subseteq TIME(n^{O(\log \log n)})$  

- $O(1)$-approx, PTAS for planes in $\mathbb{R}^3$  

  [BGK+]  
  
  [SS]  
  
  [ADM’03]
TSPN: $O(1)$-Approx for Planes in 3D  [Arkin-Demaine-M]:

$\Pi = \{\pi_1, \ldots, \pi_n\}$

Find min-radius ball, $B^*$, intersecting $\Pi$:
Let $R =$ radius of MEB(OPT); $r =$ radius of $B^*$
Then, $r \leq R$

$\ell^* =$ length of OPT

Claim: $\ell^* \geq 2R$, since OPT is contained in the ball $B(p, \ell^*)$, centered at any $p \in$OPT, of radius $\ell^*$.
Thus, $\ell^* \geq 2r$
TSPN: $O(1)$-Approx for Planes in 3D (cont):

**Algorithm:** Choose a tour, $T$, with $CH(T) \supset B^*$

In particular, we choose to traverse a Hamiltonian cycle of the edge graph (1-skeleton) of a regular simplex inscribing the ball: Length is $\gamma(d) \cdot r$

By Lemma below, $T$ visits $\Pi$

**Lemma:** If $CH(S) \cap \Pi \neq \emptyset$ and $S$ connected, then $S \cap \Pi \neq \emptyset$

Use grids: PTAS

**OPEN:** Is TSPN for planes in $\mathbb{R}^3$ NP-hard?
TSPN for Lines in 2D:

\[ S = \text{set of } n \text{ lines in } \mathbb{R}^2 \]

Watchman route in simple polygons imply poly-time algorithm \( O(n^4 \log n) \)

**OPEN:** Simple (faster) algorithm?

\[ L_1 \text{ version solvable in } O(n) \text{ time} \]

[JM]
TSPN for Lines in 3D:

Recent progress: TSPN on lines in $\mathbb{R}^3$

- NP-hard, even for lines in general position
- $O(1)$-approx (?)

OPEN: Is there a PTAS?
TSPN – More Questions:

**OPEN:** PTAS for TSPN with disjoint regions?

**OPEN:** TSPN for partitioning regions?

(e.g., faces of a triangulation)
TSPN for Disconnected Regions:

**OPEN:** $O(1)$-approx for disconnected regions (discrete point sets)

Group (Class) Steiner Tree or One-of-a-Set TSP are challenging

- $O(\log^2 n \log k)$-approx for $k$ subsets of $n$ points
- $3K/2$-approx for one-of-a-set TSP with subsets of size $\leq K$

**OPEN:** Can geometry help?
Other Variants on the Classic TSP:

- Max TSP: max tour length
  - 5/7-approx in metric spaces
  - PTAS in $\mathbb{R}^d$ for $L_p$ metrics
  - $O(n^{f-2} \log n)$ for fixed $d$
    ($f = \# \text{ facets for } \text{“disk”}$)
    e.g., $L_1$ or $L_\infty$ in $\mathbb{R}^2$: $O(n^2 \log n)$
  - $O(n)$ for $L_1$ or $L_\infty$ in $\mathbb{R}^2$
  - NP-hard for $L_2$ in $\mathbb{R}^d$, $d \geq 3$

**OPEN:** Complexity of Max TSP in Euclidean plane?

**OPEN:** Complexity of Max noncrossing TSP in Euclidean plane?
Other Variants on the Classic TSP (cont):

- bottleneck TSP: min the max edge length
  - 2-approx in metric spaces \((\text{best possible})\)
  - (no \(O(1)\)-approx without \(\Delta \neq\))
  - NP-hard in Euclidean plane

**OPEN:** Better than 2-approx in \(E^2\)?

(Ham. cycle in grid graphs)
Other Variants on the Classic TSP (cont):

- max scatter TSP: max the min edge length
  - NP-complete in metric spaces
  - 2-approx in metric spaces (best possible)
    (no $O(1)$-approx without $\Delta \neq$)

**OPEN:** Complexity of max scatter TSP in the plane?

**OPEN:** Better than 2-approx using geometry?
Other Variants on the Classic TSP (cont):

- minimum latency: “traveling repairman problem”

  Given starting point

  **Goal**: Min the sum of the *arrival times* at all other points

  NP-hard in $E^2$; 3.59-approx $\text{(10.78-approx, metric spaces)}$

  quasipoly-time approx scheme in $\mathbb{R}^2$ $O(n^{O(\log n \log \log n/\varepsilon^2)})$, [ArKa99]

  **OPEN**: PTAS for min latency in Euclidean plane?

  (metric version is MAX-SNP-hard)
Orienteering Problem: “Bank Robber Problem”

Invert the objective:

Maximize the number of sites visited, subject to a length constraint

(Dual of “quota-driven salesperson” problem)

- 2-approx in metric spaces (unrooted)
  (rooted version is still open)

**Lemma**: a c-approx for k-TSP gives 2c-approx for orienteering

- 2-approx in \( \mathbb{R}^2 \) (rooted or not) [AMN98]

**OPEN**: Better approx in Euclidean plane?
Optimal Tours with $\mathcal{D} = \text{Simple Polygon, } P$:

- **Metric**: shortest path (geodesic) metric in $P$
- **Special case**: $S = V$ 
  Then, OPT tour is the boundary, $\partial P$ 
  OPT path visiting $V$: $O(n^3)$, DP 
- **General set of point sites** $S \subset P$ 
  NP-hard (strict generalization of Euclidean TSP) 
  Recent result: PTAS 
  (method: apply $m$-guillotine subdivisions, DP) 

**Generalization**: Polygons with $O(1)$ holes

**OPEN**: PTAS for polygons with arbitrary holes?
Milling/Lawnmowing:

Objective: optimally move a cutter to sweep over a given region
A type of TSPN problem: a continuum of regions (cutter disks)

Milling: keep cutter inside the region
Mowing: cutter may leave the region

- Mowing is NP-hard for polygons (even simple)
  
  $(3 + \epsilon)$-approx

  New: PTAS

- Related: TSP of segments: PTAS

- Milling is NP-hard in polygons with holes

  OPEN: Complexity of min-length milling in simple polygons?

  2.5-approx
Milling with Turn Costs:

(with Arkin, Bender, Demaine, Fekete, Sethia)

Milling with turn costs: min-link milling
(unit square cutter, rectilinear polygons (with holes))

Results include:

- NP-completeness
- PTAS (for constant # holes) (m-guillotine method generalization)
- relatively simple 3.75-approx
Link Metric in $\mathbb{R}^2$:

Point set $S$

- Allow tour to cross itself
  - NP-hard
    (from covering points by lines)
  - $O(\log n)$-approx
    (using set cover)

- Simple tours (no self crossing)
  - OPEN: NP-hard?
  - $O(\log n)$-approx
    ($O(1)$-approx for rectilinear tours)
    (BSP, DP)
Link Metric in Polygon $P$:

Point set $S \subset P$

- Simple polygon $P$
  
  NP-hard  (even if $S \subset \partial P$)  
  
  (from exact 3-cover)

  $O(1)$-approx

- Polygon $P$ with holes

  NP-hard, even if holes are all convex

[AL93]

[AMP]
Turn Metrics:

- Angular-metric TSP: min the sum of the direction changes
  “total turn”
  NP-complete

- Angle-Restricted Tour Problem: turns angles in allowable set $A$

\[ i+1 \quad +132 \quad i-1 \]

\[ i+1 \quad i \quad i \quad i-1 \]

\[ -38 \]

\[ i+1 \]
Turn Metrics – Angle-Restricted Tour (cont):

Various results on detecting $A$-tours:

- $A = [0, \pi], n \neq 4$: $\exists A$-tour
  i.e., any $S$ admits a pseudoconvex (right-turning) tour
- $A = \{-\pi/2, \pi/2\}$: poly-time to detect if $\exists A$-tour
- $A = \{-\pi/2, \pi/2, \pi\}, \{-\pi/2, \pi\}, \{\pi, \pi/2\}$: NP-complete
- $A = (-\pi/2, \pi/2)$ (acute) or $A = (-\pi, -\pi/2) \cup (\pi/2, \pi]$ (obtuse)
  $\exists S$ arbitrarily large without $A$-tour

**OPEN:** Complexity of determining if $\exists$ acute (or obtuse) tour?
Another Nice Problem:

Given $n$ points in the plane.

**OPEN:** Does there always exist a spanning path with all angles $\geq \pi/6$?
Area Optimization:

Min-Area TSP (resp., Max-Area TSP):
find a *simple* tour on $S \subset \mathbb{R}^2$ of min (max) area

Both problems are NP-complete

- Max-Area TSP
  $(1/2)$-approx in $O(n \log n)$ time (surround $\geq \frac{1}{2}$ area of $CH(S)$)
  (NP-complete to determine if $> \frac{2}{3} + \epsilon$ of area can be obtained)

- Min-Area TSP

  **OPEN:** Is there a poly-time approx algorithm for Min-Area TSP?

  (none for min-area disjoint triangle matching on $3n$ points)
Freeze Tag Problem:

Given: $n$ robots at points in a metric space

$n - 1$ robots are “asleep”

1 robot is awake

An awake robot “wakes up” a sleeping robot at point $p$ by going to $p$

As robots wake up, there are more robots to assist in waking up others

**Goal:** Wake up all robots as soon as possible

Minimize makespan
Other Motivations:

- Distribute data (or other commodity) to agents, where physical proximity is necessary for transmittal
- Secret sharing by whisper
- Natural network optimization problem:

  Minimize the length of a root-to-leaf path in a *binary* spanning tree
  Minimum-diameter bounded-degree spanning tree.

Related open question:

**OPEN:** Is there a PTAS for min-weight bounded-degree spanning tree?

(degree bounds of 3, 4 are relevant in $\mathbb{R}^2$)
Related Work:

Dissemination of data in graphs:

- minimum broadcast time problem
- multicast problem
- minimum gossip time problem

Key differences from FTP:
- messages sent along edges of graph (no need for proximity)
- broadcast problem poly in trees, but FTP is NP-hard even for stars
Simple Approximation Bounds:

Any “brain-dead” strategy gives $O(\log n)$-approx:

- Source robot awakens one other robot
  (travelling distance $\leq \text{D}$, diameter)
  Now 2 awake robots.

- Each travels to a distinct other asleep robot (dist $\leq \text{D}$), awakens it and waits
  (if necessary) for the other robot to reach its destination
  Now 4 awake robots.

- Etc, etc, $\log n$ rounds, each of length $\leq \text{D}$

- Lower bound on OPT: $t^* \geq R_0 = \max_{p_i} d(v_0, p_i) \geq \frac{\text{D}}{2}$

OPEN: Is there an $o(\log n)$ approximation algorithm?
Fundamental Question:

Whether to awaken a nearby robot or go further to (start to) awaken a larger swarm?

Close individual

Further away cluster
Summary of Results:

1. FTP is NP-hard, even for stars, with one robot per leaf
2. $O(1)$-approx for (general) stars
3. PTAS for stars, same number of robots at each leaf
4. Tight analysis of greedy heuristic on stars: $7/3$-approx
5. $o(\log n)$-approx for FTP in ultrametrics
   ($2^{O(\sqrt{\log \log n})}$-approx)
6. Simple linear-time on-line algorithm, $O(\log \Delta)$-competitive
   ($\Delta \leq \text{max degree}$)
7. NP-hard to get $<5/3$-approx in offline problem, even if $\Delta \leq 5$
8. PTAS for geometric instances in fixed-dimension, $L_p$ metric
   Time $O(n \log n + 2^{poly(1/\varepsilon)})$
Summary of Results (Continued):

9. Greedy heuristic applied to geometric instances gives an $O((\log n)^{1-1/d})$-approx in $d$ dimensions.
   
   $d = 1$: $O(1)$-approximation
   $d = 2$: $O(\sqrt{\log n})$-approximation.
   
   We prove that this analysis is tight by supplying matching lower bounds.

10. Experimental investigation of heuristic strategies for the FTP, comparing the different choices of greedy strategies and comparing these greedy strategies with other heuristics.

11. “Density-Based” strategy: simple $O(1)$-approx for Euclidean spaces and an $O((\log n)^{\log(5/3)})$-approx for unweighted graphs.
12. “Sibling-Based” strategy that yields:

1. An $O(1)$-approximation for unweighted graphs.
2. An $O(\sqrt{\log n})$-approx, if unweighted edges, weighted nodes.
3. An analysis of the makespan when the strategy is applied to general graphs, having weighted edges and weighted nodes. We show that the makespan is $O(d + L \log n)$, resulting in an $O((L/d) \log n)$-approximation algorithm. ($L =$ length of longest edge)
4. An $O(1)$-approx for minimizing total distance traveled by all robots in a general tree.

13. We prove that the FTP is NP-hard in unweighted graphs.
Geometric Instances:

Robots at points in the plane:

Question: Can we exploit geometry to get good approx?

OPEN: Is the problem NP-hard?
Geometric Instances: $O(1)$-approx:

**Theorem:** \( \exists \) an $O(1)$-approx, time $O(n \log n)$, for the geometric FTP in any fixed dimension $d$. The algorithm gives wake-up schedule with makespan $O(\text{diam}(R))$.

**Strategy:** When robot at $p$ awakens, it awakens nearest asleep robot in each of $K$ sectors, in order of increasing distance from $p$. 
• $G_K = (R, E_K)$ is a $\Theta$-graph (which is a $t$-spanner)

• Distances in $G_K$ approx Euclidean

• Let $v_\ell$ be the last robot to be awakened

• If robot at point $v$ is awakened at time $t$, then all neighbors of $v$ in $G_K$ are awakened by time $t + \xi$, where $\xi = \text{length of path } v, u_1, u_2, \ldots, u_j$.

• $\xi \leq (2j - 1) \cdot d(v, u_j) \leq (2K - 1) \cdot d(v, u_j)$

• The path from $v_0$ to $v_\ell$ in the wake-up tree has length at most $(2K - 1) \cdot d_{G_K}(v_0, v_\ell) \leq O(1) \cdot d(v_0, v_\ell)$
PTAS for Geometric Instances:

Rescale so that all robots lie in unit square
Look at $m$-by-$m$ grid of pixels, $m = O(1/\epsilon)$

Consider an enumeration over a special class of wake-up trees on a set $P$ of representative points, one per occupied pixel
A wake-up tree is *pseudo-balanced* if each root-to-leaf path has $O(\log^2 n)$ nodes.

**ALGORITHM:**

0. Pick a representative point in each occupied pixel → set $P$

1. Among all pseudo-balanced wake-up trees for $P$, pick one ($T_b^*(P)$) of min makespan, $t_b^*(P)$
   (outdegree at most $\min\{m^2 - 1, \ell + 1\}$ if $\ell$ robots in pixel)
   Only $2^{O(m^2 \log m)}$ trees.

2. Convert $T_b^*(P)$ into a wake-up tree for *all* robots by replacing each $p \in P$ with an $O(1)$-approx wake-up tree for robots in $p$’s pixel
   (time $O(n \log n)$)

**Total time:** $O(2^{O(m^2 \log m)} + n \log n)$
Correctness:

**Lemma 1.** There is a choice of representative points $P$ such that the makespan of an optimal wake-up tree of $P$ is at most $t^*(R)$.

**Proof:** Just pick the representative point to be the location of the first robot that is awakened in an opt solution, $T^*(R)$, for the set $R$ of all robots.
Lemma 2. If \( \exists \) wake-up tree, \( \mathcal{T} \), of makespan \( t \), then, for any \( \mu > 0 \), there exists a pseudo-balanced awakening tree, \( \mathcal{T}_b \), of makespan \( t_b \leq (1 + \mu)t \).

Proof Sketch:

Use a heavy path decomposition of \( \mathcal{T} \)

Any root-to-leaf path has only \( O(\log n) \) light edges

Decompose each heavy path into short subpaths of length \( \xi = \mu t / \log n \)

(only \( O((1 + 1/\mu) \log n) \) subpaths on any root-to-leaf path of \( \mathcal{T} \))

Modify wake-up tree \( \mathcal{T} \) to transform each subpath into a wake-up tree of height \( O(\log n) \), with a small increase in makespan
Lemma 3. For any two choices, $P$ and $P'$, of the set of representative points, $t^*_b(P) \leq t^*_b(P') + O((\log^2 m)/m)$.

Proof: Pixels have size $O(1/m)$ and there are at most $O(\log^2 m)$ awakenings in each root-to-leaf path of a pseudo-balanced tree; thus, any additional wake-up cost is bounded by $O((\log^2 m)/m)$. 
Lemma 4. For any pseudo-balanced wake-up tree of $P$, there exists a wake-up tree, $T(R)$, with makespan $t(R) \leq t_b(P) + O((\log^2 m)/m)$. 
Putting the Pieces Together:

**Theorem:** There is a PTAS, with running time $O(2^{O(m^2 \log m)} + n \log n)$, for the geometric FTP in any fixed dimension $d$.

**Proof:**
The makespan, $t$, of the wake-up tree we compute obeys:

\[
\begin{align*}
t & \leq t_b^*(P) + O((\log^2 m)/m) \\
& \leq t_b^*(P') + 2 \cdot O((\log^2 m)/m) \\
& \leq (1 + \mu)t^* + O((\log^2 m)/m) \\
& \leq t^* \left(1 + \mu + \frac{C \log^2 m}{m}\right) \\
& \leq t^*(1 + \epsilon),
\end{align*}
\]

for appropriate choices of $\mu$ and $m$, depending on $\epsilon$. 