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 $t$. 
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 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

last-step SPM
**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:**

![Diagram showing disjoint convex polygons with paths from $s$ to $t$.]
Unconstrained TPP: Disjoint Convex Polygons:

Lemma: For any $t \in \mathbb{R}^2$ and any $i \in \{0, \ldots, k\}$, there exists a 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 $\pi_i(p) = \pi_{i-1}(p') \cup \overline{p'p}$. 
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 $\Omega((n - k)2^k)$
**Good news**: worst-case size cannot be *bigger* than “huge”:

**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**:

**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 \text{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 $S_i$ \([O(\log |P_i|)]\)

- cell $\sigma$ rooted at vertex $v$ of $T_i$
  
  $\longrightarrow$ 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 $r^b_i(v_j)$ and $r^s_i(v_j)$ 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.
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 Problem:

What is the complexity of the TPP for disjoint non-convex simple polygons?
3D Shortest Paths: Background:

- NP-hard in general [CR]

- FPTAS [Pa],[Cl],[CSY],[H-P]

  \((1 + \epsilon)\)-approx in time poly\((n, 1/\epsilon)\)

- Special cases: surfaces, \(k\) convex polytopes, buildings of \(k\) heights

  (time \(O(n^{O(k)})\))
Shortest Paths Among Stacked (Flat) Obstacles:

If obstacles are **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 \textit{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.
Splitter gadget
Shuffle gadget
Blocker

Clause filter
Literal filter
3-way splitter
Path Splitting Gadget:
Path Splitting Gadget:
Path Splitting Gadget:
Path Shuffle Gadget:
Path Shuffle Gadget:
Path Shuffle Gadget:
Path Shuffle Gadget:
Path Shuffle Gadget:
A Blocker:
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?