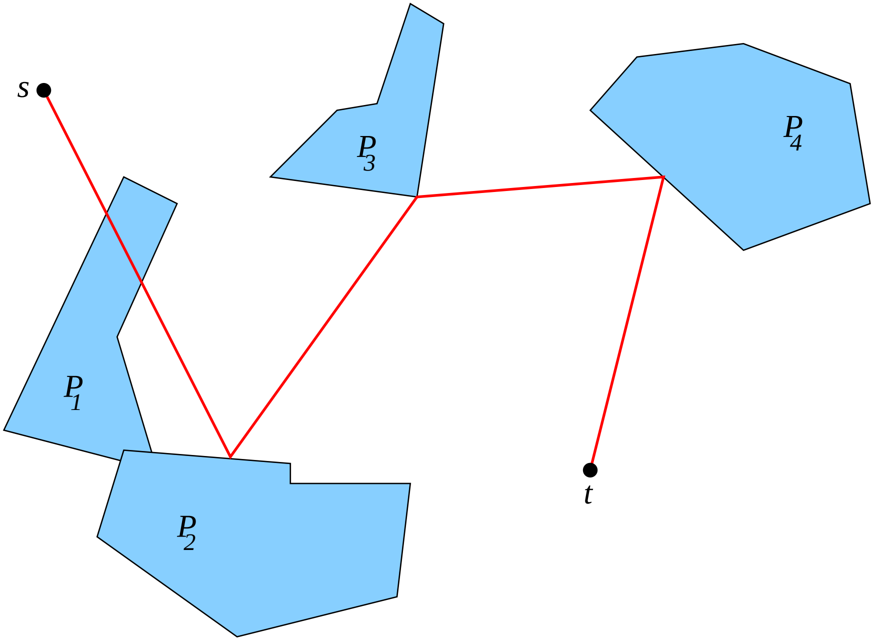
Touring a Sequence of Polygons

Moshe Dror[1] **Alon Efrat**[1] **Anna Lubiw**[2] **Joe Mitchell**[3]

[1]University of Arizona
[2]University of Waterloo
[3]Stony Brook University

Problem:

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.



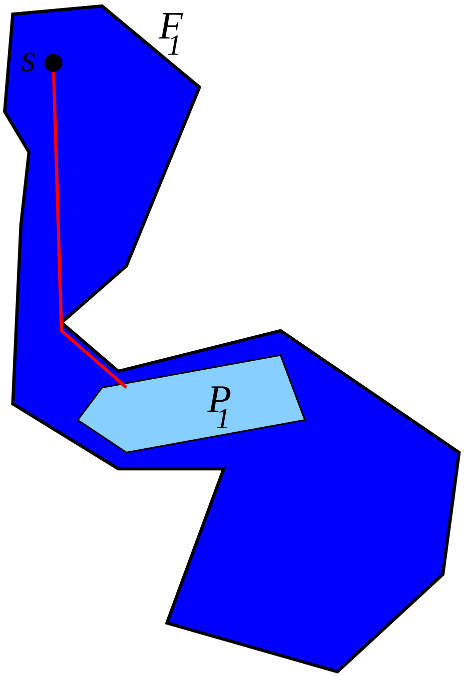
Related Problem: TSPN:

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

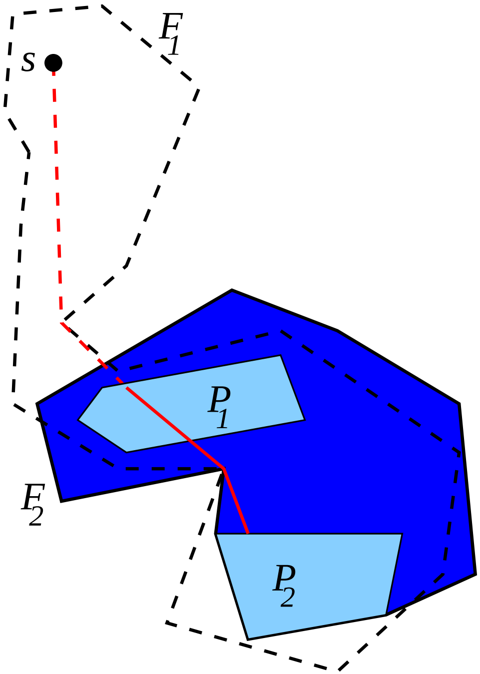
TSPN: $O(\log n)$-approx in general

$O(1)$-approx, PTAS in special cases

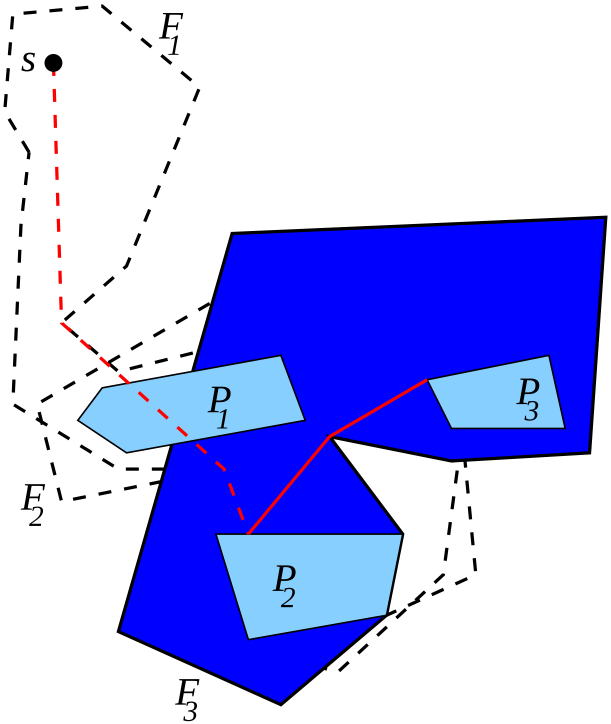
The Fenced Problem:



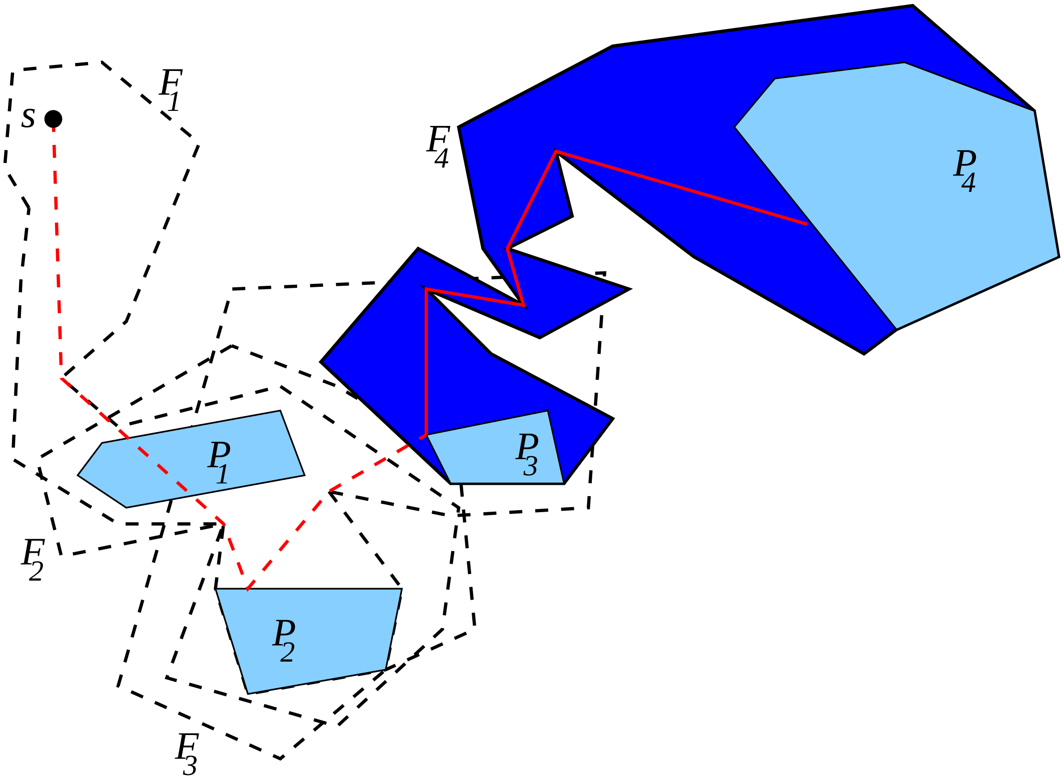
The Fenced Problem:



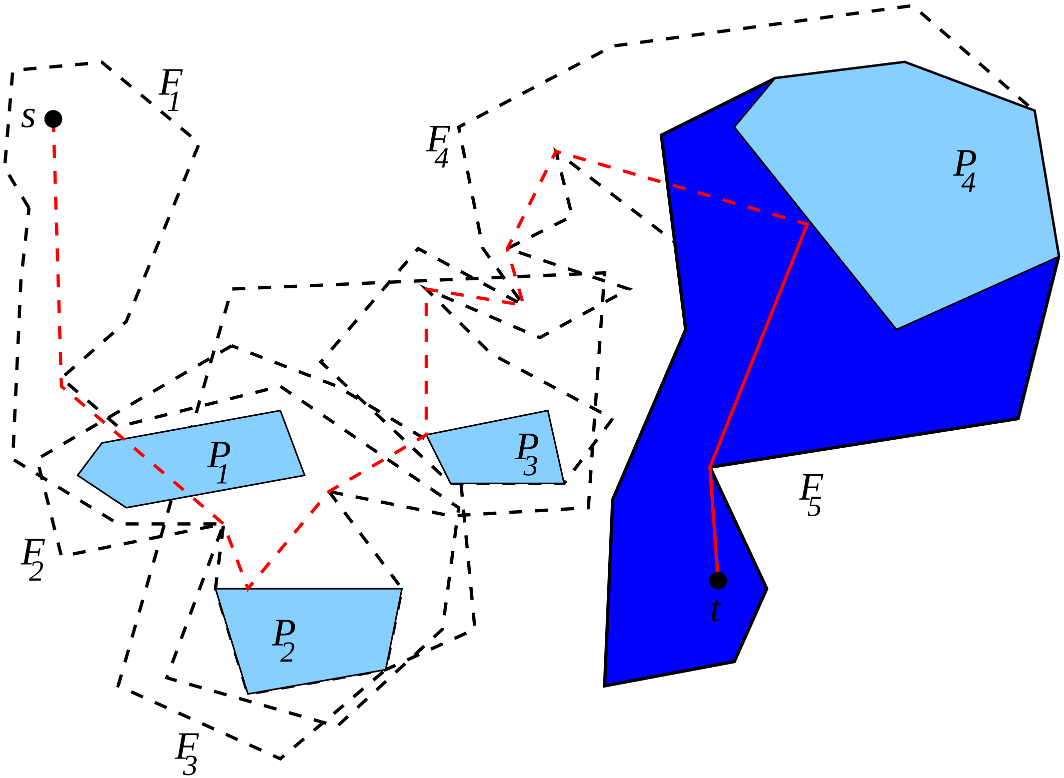
The Fenced Problem:



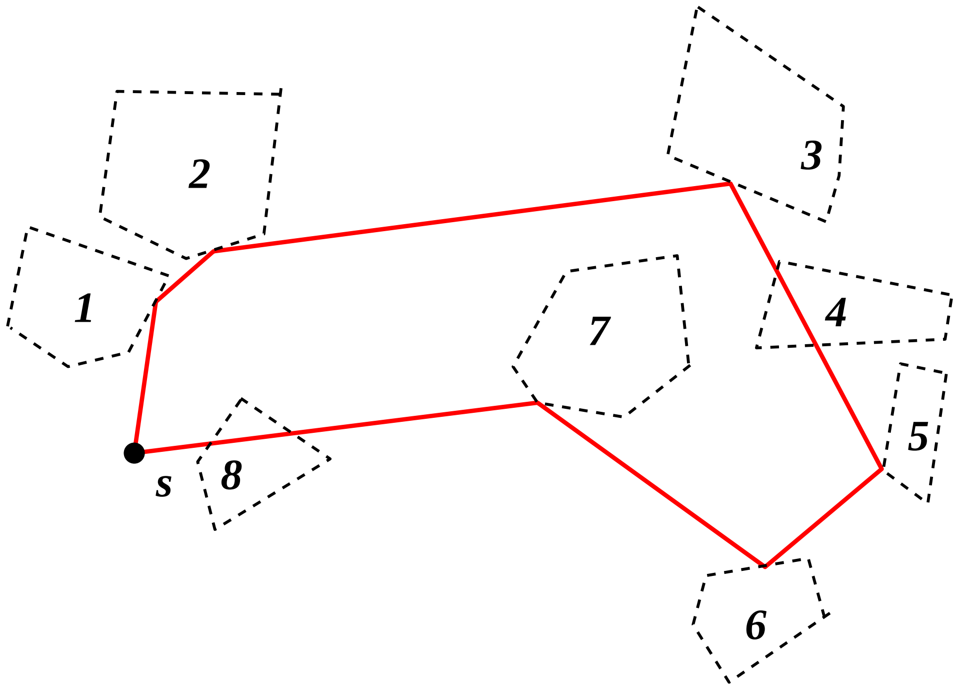
The Fenced Problem:



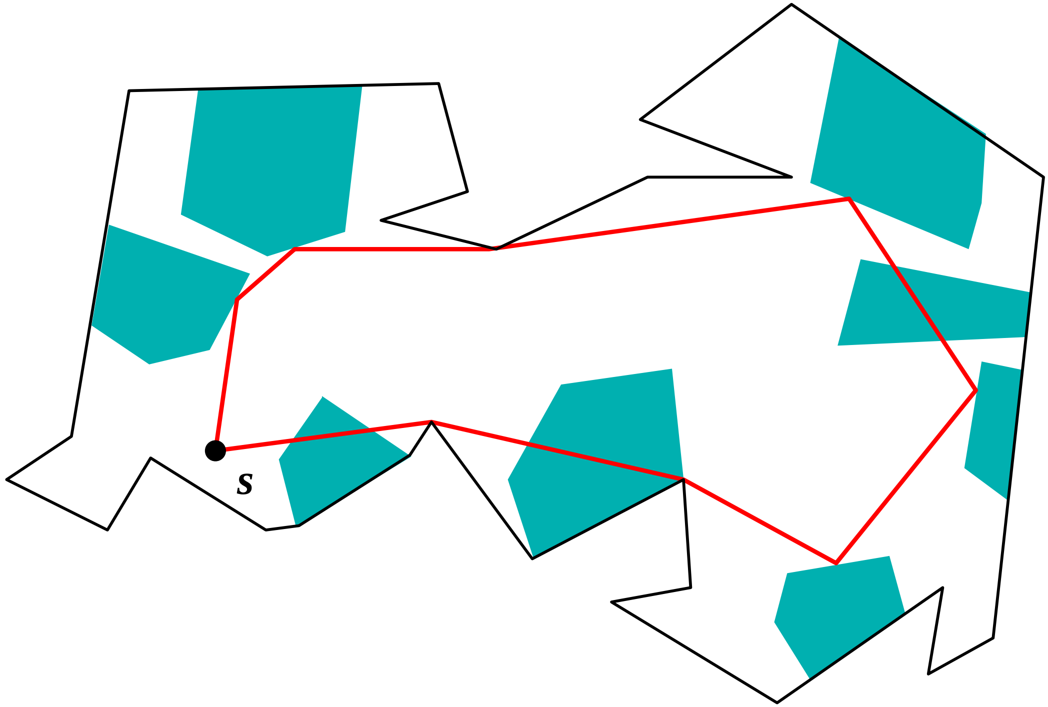
The Fenced Problem:



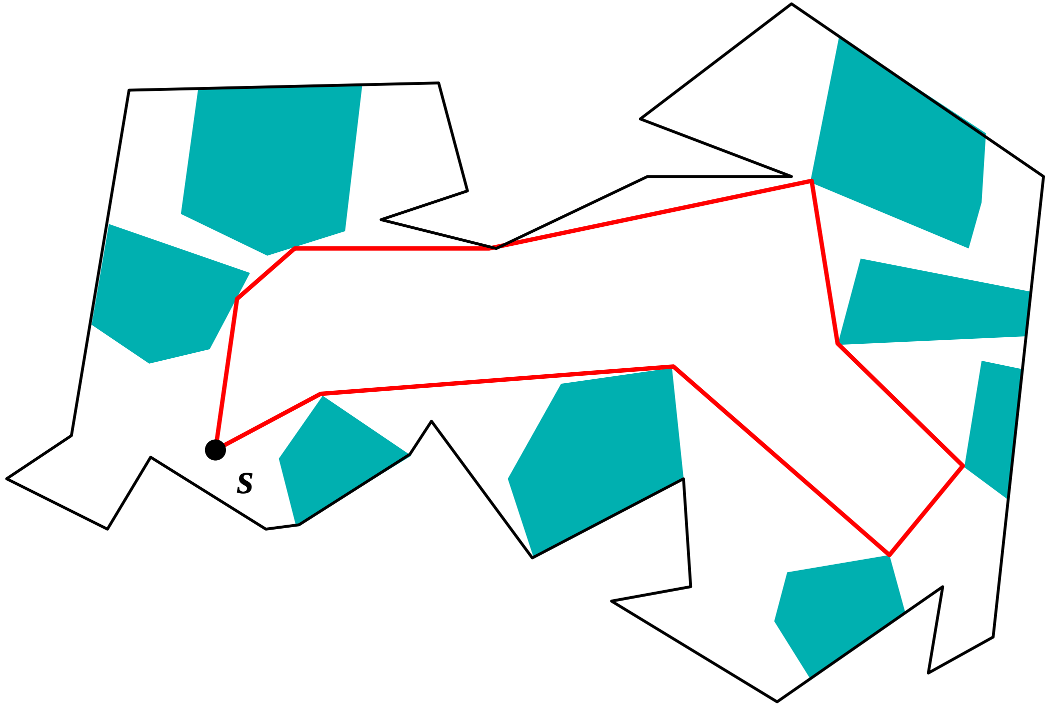
Applications: Parts Cutting Problem:



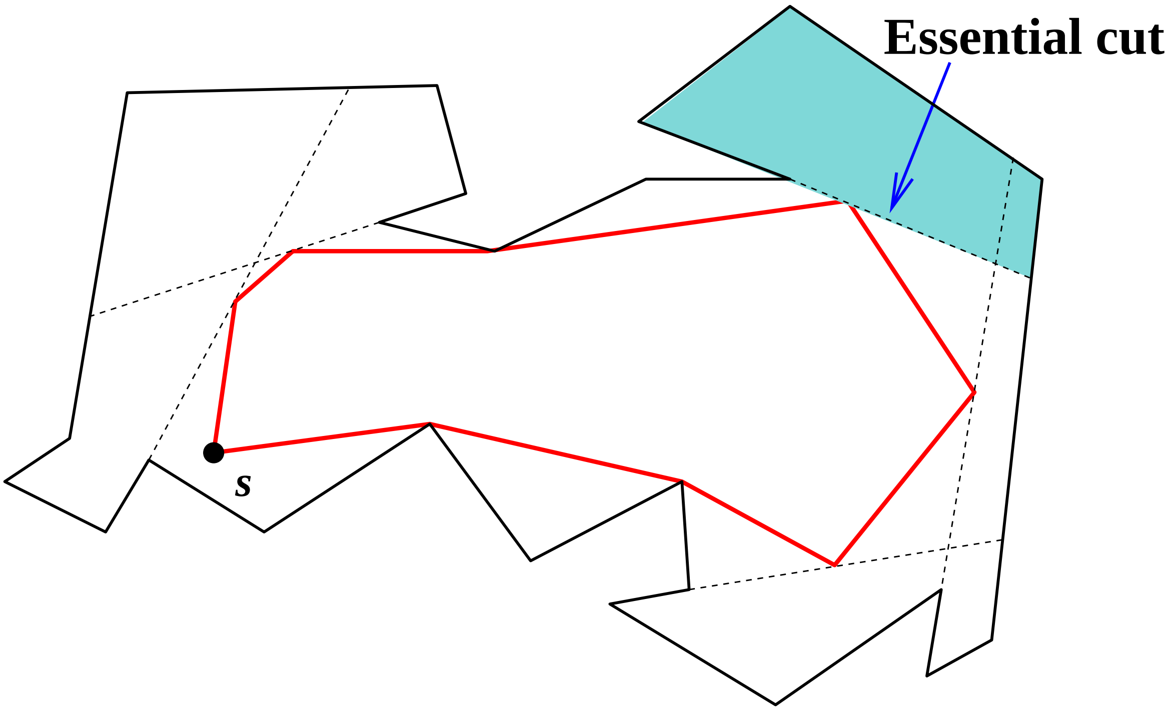
Applications: Safari Problem:



Applications: Zookeeper Problem:



Applications: Watchman Route Problem:



Summary of Results:

- Disjoint convex polygons: $O(kn \log(n/k))$ time, $O(n)$ space
  (For fixed s, $O(k \log(n/k))$ 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 shortst 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. $O(n^5)$

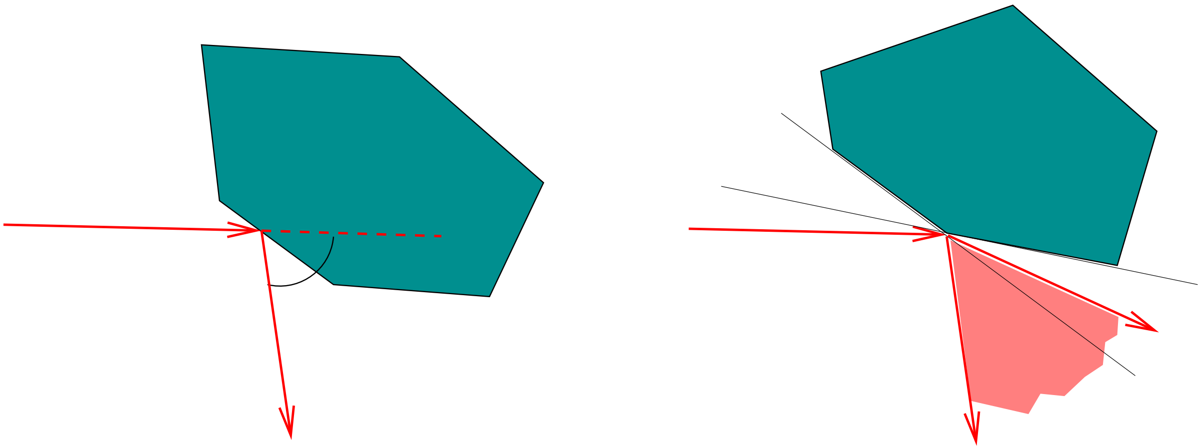
    We avoid use of complicated path "adjustments" arguments, DP
  - Parts cutting: $O(kn \log(n/k))$

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 $t \in \Re^2$ and any $i \in \{0, \ldots, k\}$, $\exists$ unique shortest i-path, $\pi_i(p)$, from $s = P_0$ to t.

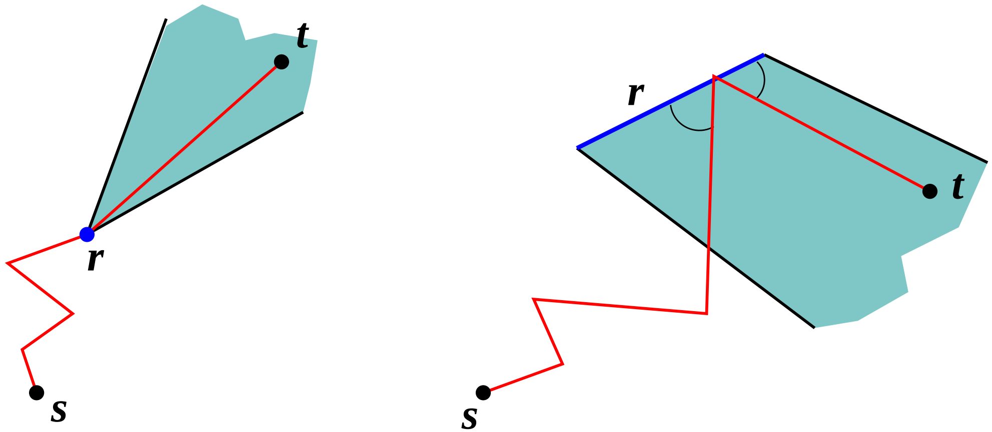
Thus, local optimality is equivalent to global optimality.

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

Lemma: For any $p \in \Re^2$ and any i, there is a unique point $p' \in T_i$ such that $\pi_i(p) = \pi_{i-1}(p') \cup \overline{p'p}$.

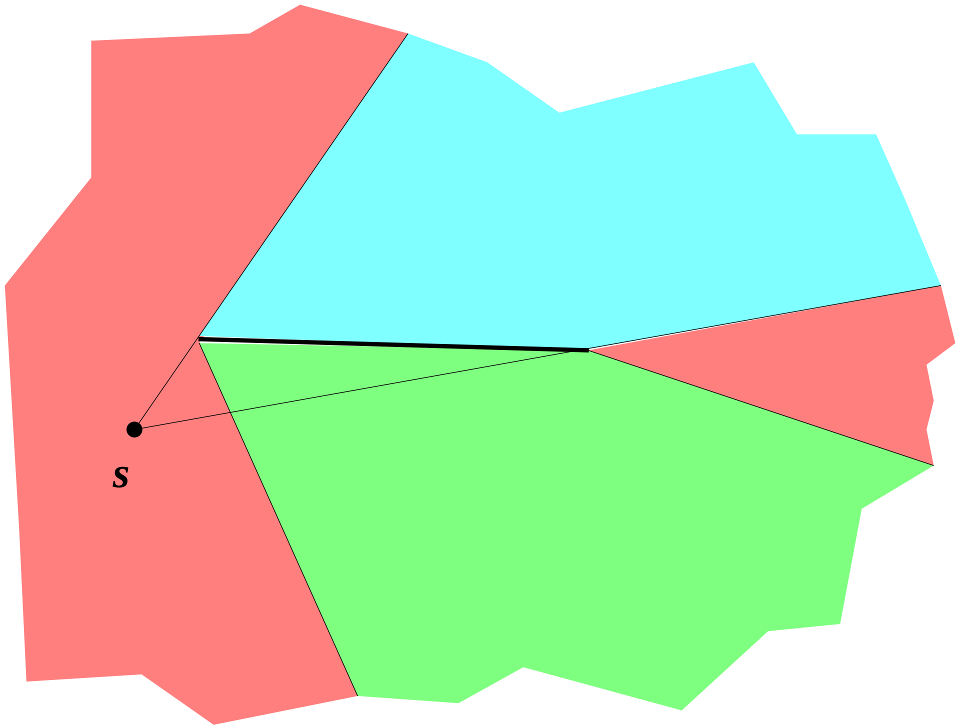
General Approach: Build a Shortest Path Map:

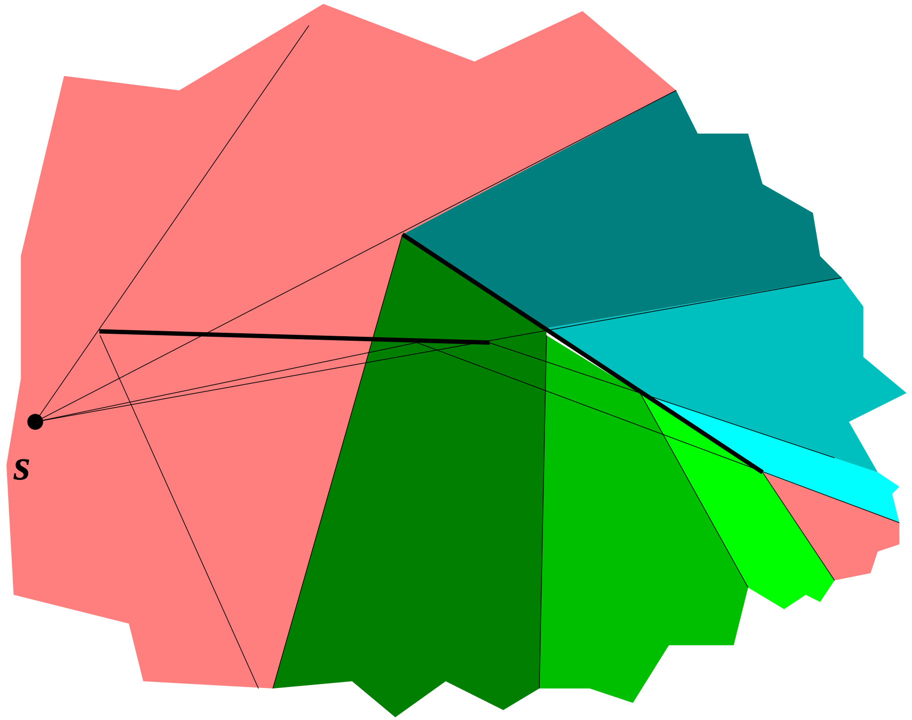
$\text{SPM}_k(s)$: a decomposition of the plane into cells according to the combinatorial type of a shortest k-path to t

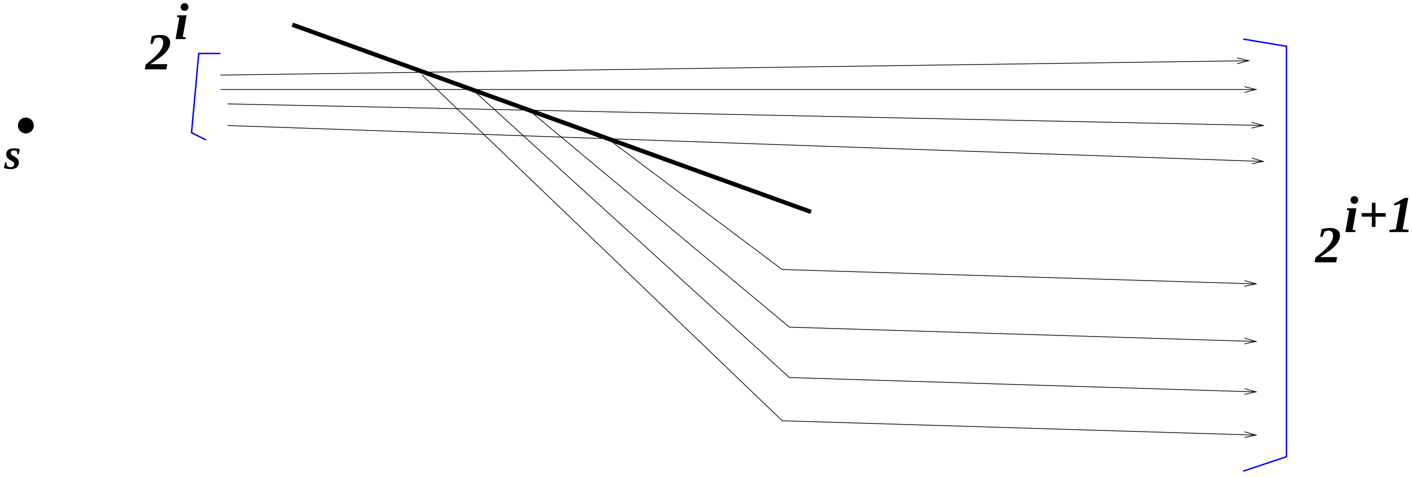


Bad news: worst-case size can be **huge**:

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


s


s



2^i
s
2^{i+1}

Good news: worst-case size cannot be *bigger* than “huge”:

Theorem: The worst-case complexity of $\mathrm{SPM}_k(s)$ 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 $\mathrm{SPM}_k(s)$ in time $O(k \cdot |\mathrm{SPM}_k(s)|)$, 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 = *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 $p \in T_i$:

$r_i^s(p)$ = set of rays of locally shortest i-paths going straight through p:

a single ray

$r_i^b(p)$ = set of rays of locally shortest i-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)

$r_i(p) = r_i^s(p) \cup r_i^b(p)$

$R_i = \cup_{p \in T_i} r_i(p)$ (an infinite family of rays) is the starburst with source T_i

The Last Step Shortest Path Map:

$\mathcal{S}_i =$ the last step shortest path map, subdivision according to the *combinatorial type* of the rays of R_i passing through points $p \in \Re^2$

$\mathcal{S}_i$ decomposes the plane into cells σ 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 σ

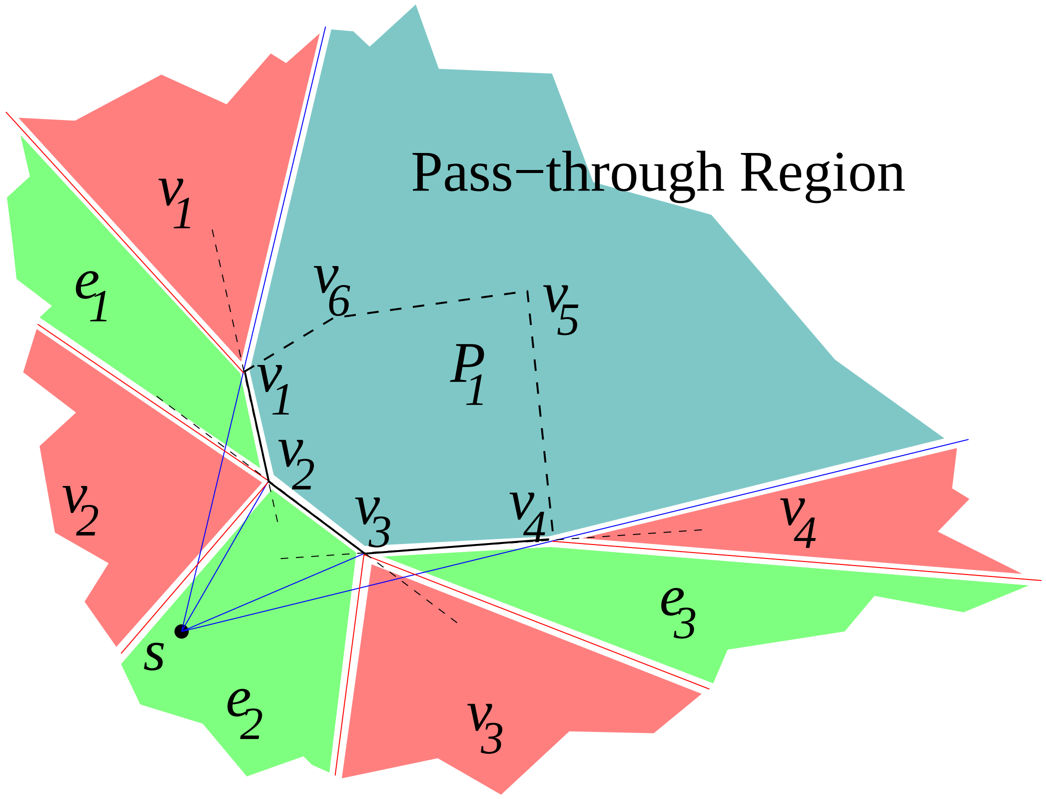
(2) unbounded 3-sided regions associated with edge e of T_i, classified as

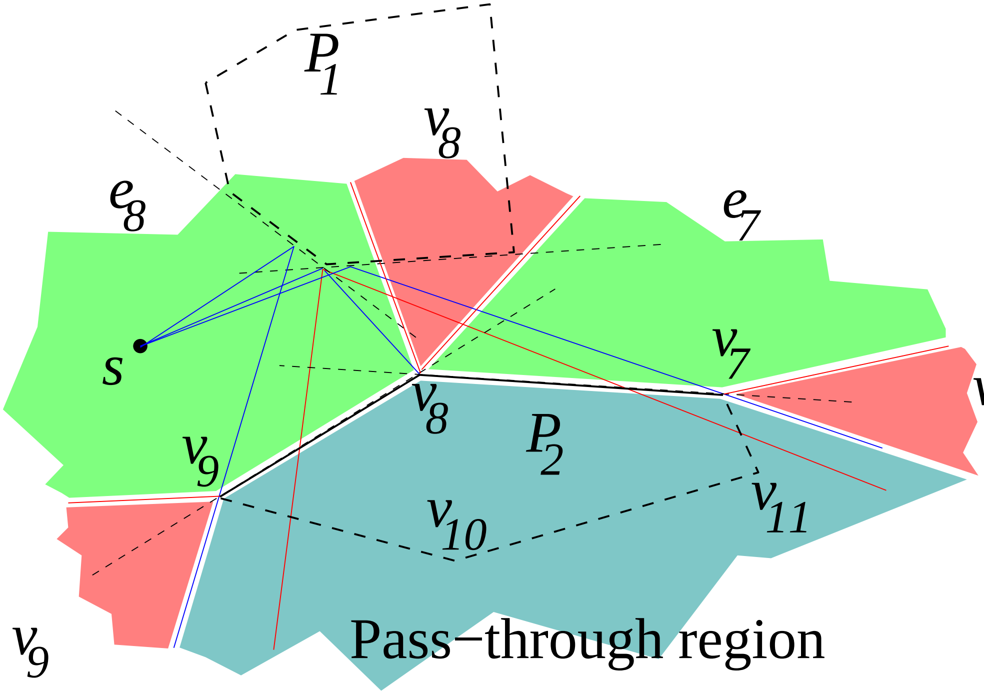
- *reflection cells* or
- *pass-through cells*

e is the source of cell σ

The pass-through region is the union of all pass-through cells

Last Step Shortest Path Map:





P_1
v_8
e_8
e_7
v_7
s
v_8
v_7
P_2
v_9
v_{10}
v_{11}
v_9
Pass−through region

Using the Last Step Shortest Path Map:

Find a shortest i-path to query point q:

Locate q in $\mathcal{S}_i$ $[O(\log |P_i|)]$

- cell σ 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 $\mathcal{S}_{i-1}$, etc)

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

Lemma: Given $\mathcal{S}_1, \ldots, \mathcal{S}_i$, $\pi_i(q)$ can be determined in time $O(k \log(n/k))$

Algorithm:

Construct each of the subdivisions $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k$ 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 $\mathcal{S}_{i+1}$.

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.

Thus, Alon gets to try to describe them....;-)

Needed for Safari, Watchman Route, Zookeeper.

TPP on Nonconvex Polygons:

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.

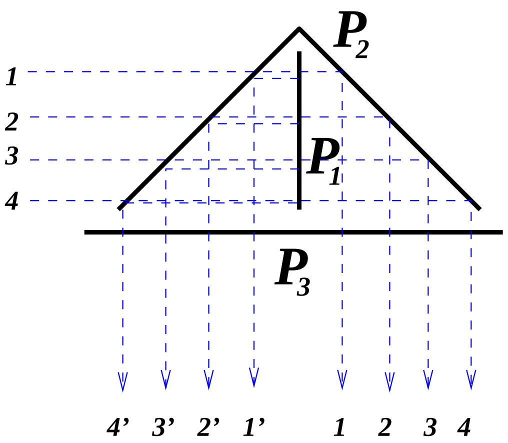
TPP on Nonconvex Polygons:

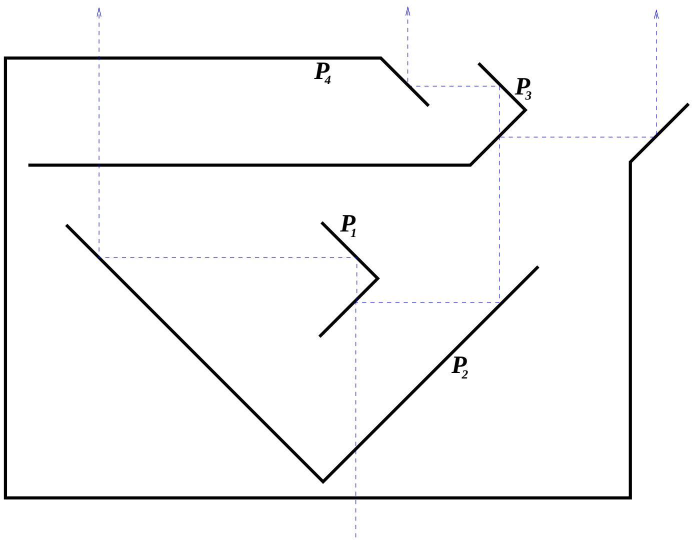
Theorem: The touring polygons problem is NP-hard, for any L_p metric ($p \geq 1$), in the case of nonconvex polygons P_i, even in the unconstrained ($F_i = \Re^2$) case with obstacles bounded by edges having angles 0, 45, or 90 degrees with respect to the x-axis.

Proof: from 3-SAT

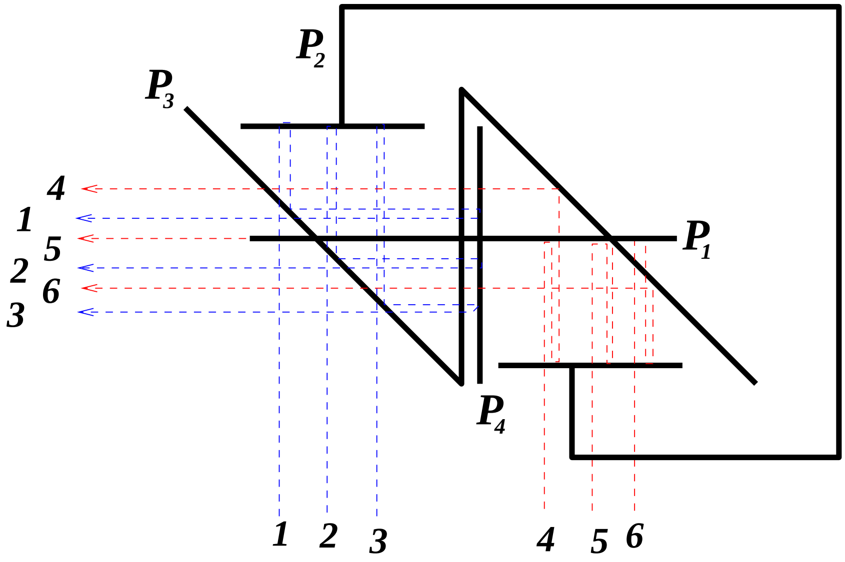
based on a careful adaptation of Canny-Reif proof

Splitter Gadgets:

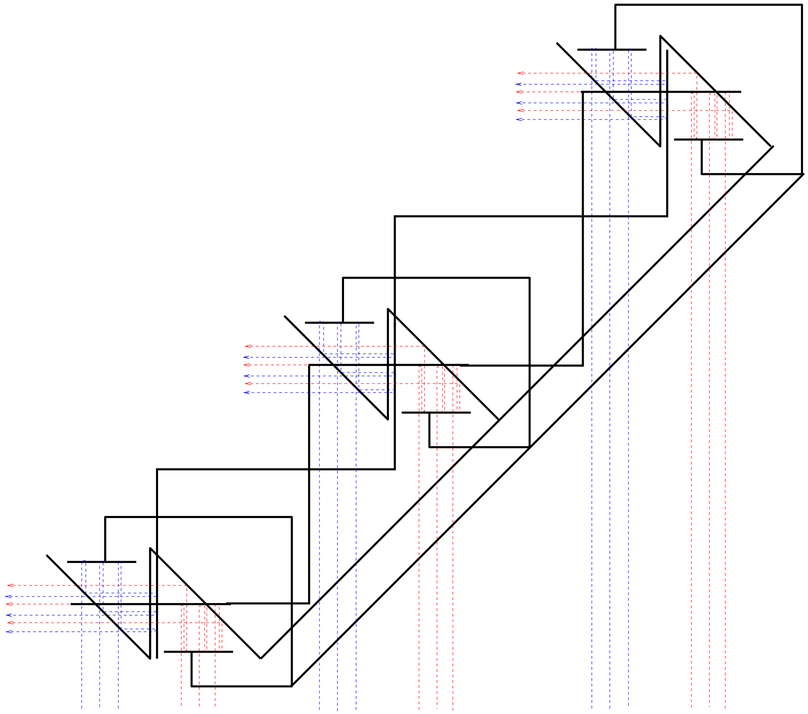




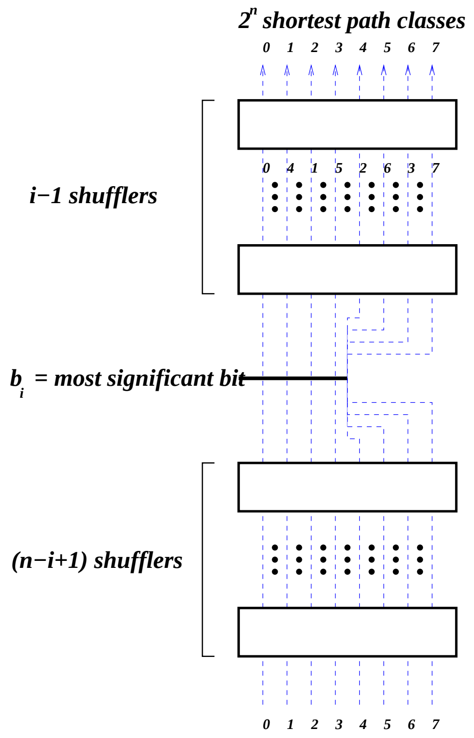
Shuffle Gadget:



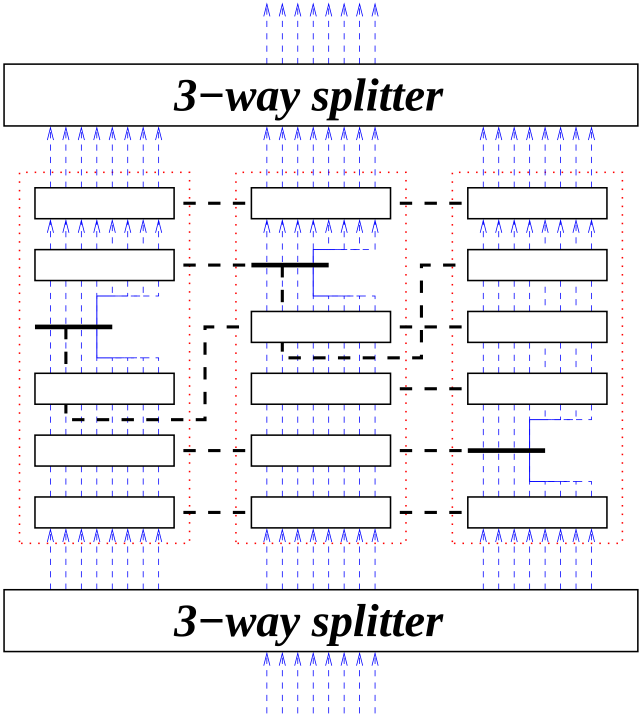
Ganging Three Shuffle Gadgets:



Literal Filter:



Clause Filter:



Open Problem:

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