UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

O Problema da Visita de Polígonos

Gabriel Freire Ushijima

São Paulo, SP
24 de setembro de 2025

1 Introdução

Este relatório descreve a implementação e os resultados obtidos na resolução do problema da visita de polígonos, usando como base o paper de Mitchell [1] que descreve algoritmos para o caso sem e com restrições. Buscamos apresentar uma abordagem mais prática e detalhada para o problema, sem um foco tão grande na análise teórica.

2 O Problema de Visita de Polígonos Irrestrito

Vamos tomar como base a implementação do algoritmo de Mitchell para o problema irrestrito que fizemos em *Python*. O código pode ser encontrado no arquivo `TouringPolygons/problem1.py`.

2.1 Definições e Notação

Considere o seguinte problema: dados dois pontos $s, t \in \mathbb{R}^2$ e uma sequência de polígonos convexos disjuntos $P_1, P_2, \ldots, P_k$, encontrar o caminho de menor comprimento que se inicia em s, termina em t e toca cada polígono P_i em pelo menos um ponto, podendo atravessá-los.

A figura ao lado ilustra um exemplo de entrada e a solução ótima para o problema para um caso com 3 polígonos. Temos s como o ponto verde, t como o ponto vermelho e os polígonos P_1, P_2 e P_3 como o triângulo azul, o trapézio laranja e o pentágono verde, respectivamente. O caminho mínimo é representado pela linha roxa.

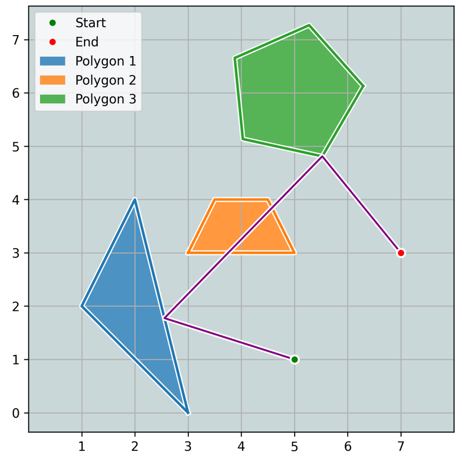

Figura 1: Caminho mínimo para um caso de 3 polígonos.

Retomando o paper, definimos um i-path até p como um caminho mínimo que começa em s, encosta em cada um dos polígonos $P_1, P_2, \ldots, P_i$ e termina em p. Note que nosso objetivo é encontrar um k-path até t. Enquanto não vamos entrar em detalhes, todo i-path é único.

A função central desse algoritmo será a função Query(p, i), que recebe um ponto p e um índice i e retorna o penúltimo ponto q do i-path até p (ou seja, o ponto imediatamente anterior a p nesse caminho). Note que a resposta do problema pode ser obtida chamando Query(t, k).

Primeiramente, vamos descrever procedimentos auxiliares que serão úteis para a implementação da função Query. Finalmente, vamos descrever como respoder consultas usando esses procedimentos. Por enquanto, vamos assumir que sabemos como respoder as consultas.

2.2 Particionando o Plano

O primeiro passo do algoritmo é, para cada polígono P_i, criar uma partição S_i do plano. Essa partição tem 3 tipos de regiões, Regiões de Vértice, Regiões de Aresta e Regiões de Atravessa. Essa partição é usada para responder consultas, uma vez que o comportamento da função Query(p, i) depende de qual região p pertence.

- Regiões de Vértice.
- Regiões de Aresta.
- Regiões de Atravessa.

Referências

[1] Moshe Dror, Alon Efrat, Anna Lubiw, and Joseph S. B. Mitchell, *Touring a sequence of polygons*, STOC ’03 (2003), 473 –482.