



Universidade Federal da Paraíba
Centro de Informática
Lógica Aplicada à Computação

Problema do Caixeiro Viajante em PROLOG

Alunos:

Isabella Nascimento de Sousa	Mat: 20230102395
Rodrigo Lanes Meneses	Mat: 20210025243
Vito Elias	Mat: 20210094483

Professora:

Tatiana Araújo Simões

17 de setembro de 2025

Conteúdo

1	Introdução	2
1.1	Contextualização	2
1.2	Definições	2
2	Desenvolvimento	2
3	Métodos	2
4	Resultados e Conclusão	3
	Referências	3

1 Introdução

1.1 Contextualização

O Problema do Caixeiro Viajante (Traveling Salesman Problem, TSP) é um dos mais famosos problemas de otimização combinatória na literatura. O TSP consiste em encontrar uma rota entre um conjunto de cidades que minimize distância total percorrida ou tempo total de viagem, de modo a visitar todas as cidades sem passar mais de uma vez em uma mesma cidade.

Para encontrar a solução para o problema, examinar todas as rotas possíveis em busca da menor distância percorrida torna-se inviável à medida que a quantidade de cidades cresce, pois o número de novas conexões a cada introdução de uma cidade ao grafo aumenta de maneira fatorial.

Para evitar a solução que requer elevado custo computacional, a abordagem ao problema pode ser feita por otimização combinatória, com métodos exatos e/ou métodos heurísticos.

1.2 Definições

O problema trata as cidades como nós em um grafo. Se o trajeto entre duas cidades existe, esse trajeto é representado por uma aresta que une os nós no grafo e a distância entre as duas cidades é representada pelo custo da aresta.

2 Desenvolvimento

A abordagem escolhida para a implementação em PROLOG utiliza o método exato de força bruta, buscando todas as soluções possíveis e comparando para identificar a menos custosa.

Apesar de nos dar um resultado comprovadamente ótimo, o método é limitado a rotas com poucas cidade, visto que a complexidade do problema será de $O(n!)$.

3 Métodos

Primeiramente, um código em Python funcional e que resolve o TSP por força bruta foi implementado e testado para se ter uma base e um comparativo se o PROLOG estaria nos dando o resultado corretamente. Após isso, foi desenvolvido o código em PROLOG, que foi testado com os mesmos exemplos do código Python.

4 Resultados e Conclusão

O código em PROLOG obteve os mesmos resultados que o código Python, comprovando que a implementação está correta.

No entanto, o PROLOG fica limitado a instâncias com um número máximo de 8 nós no grafo, caso contrário, ocorre um overflow de memória. Vale observar que o computador em que o código foi executado possui 32 GB de memória RAM. Possivelmente o PROLOG aceitaria mais nós se a memória fosse maior.

Conclui-se então que é perfeitamente possível resolver o problema do caixeiro viajante de maneira exata com força bruta utilizando PROLOG, mas com certas limitações relacionadas exclusivamente ao método utilizado e não à linguagem de programação em si.

Referências

- [1] Otimização combinatória: apostila do LOG-UEPB.
- [2] <https://github.com/RoyCal/TSP-PROLOG>