

O presente projecto irá desenrolar-se no contexto do clássico problema do caixeiro viajante (*Travelling Salesman Problem* – TSP):

- dado um conjunto de cidades
- parte de uma cidade origem
- passa pelas restantes cidades uma única vez
- retorne à cidade origem no final do percurso
- percorra a menor distância possível

O ficheiro `bc_tp2`, contém informação sobre a localização, baseada em latitude e longitude de um conjunto de 50 cidades. Para além disso no ficheiro têm ainda disponível o predicado *distance*, que dadas duas cidades quaisquer calcula a distância entre elas. Considere ainda que as cidades estão todas ligadas entre si.

Iteração 1

1. No ficheiro `bc_tp2` comente os factos do tipo `city` de forma a ficar apenas com 5 cidades. Implemente um predicado *tsp1*, que seja capaz de determinar o circuito de menor comprimento que dada uma cidade *C*, de entre um conjunto de outras cidades, permita visitar cada uma das cidades uma única vez e voltar à cidade inicial. Na implementação do predicado deve seguir uma abordagem *força bruta* (pesquisa exaustiva).
2. Identifique qual o número máximo de cidades que o predicado anterior tem capacidade para resolver.

Iteração 2

3. Implemente o predicado *tsp2*, utilizando uma heurística *greedy*, a **heurística do vizinho mais próximo** (ideia base: próxima cidade a ser visitada é a mais próxima que ainda não foi visitada).

Iteração 3

4. Implemente um predicado *tsp3*, que com base na solução encontrada na questão 3, implemente uma heurística de melhoria da solução que tem por base o princípio da remoção de cruzamentos.

Iteração 4

5. Implemente o predicado *tsp4*, recorrendo a algoritmos genéticos.