



**Universidade do Minho**  
Escola de Engenharia

# **Sistemas de Representação de Conhecimento e Raciocínio**

**Trabalho Individual**



**Luís Enes Sousa A89597**

8 de junho de 2021

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Dataset</b>	<b>2</b>
2.1	parser.py . . . . .	2
<b>3</b>	<b>Formulação do Problema</b>	<b>2</b>
<b>4</b>	<b>Estratégias de Procura</b>	<b>2</b>
4.1	Procura Não Informada . . . . .	2
4.1.1	Profundidade (DFS - Depth-First Search) . . . . .	2
4.1.2	Largura (BFS - Breadth-First Search . . . . .	3
4.1.3	Busca Iterativa Limitada em Profundidade . . . . .	3
4.2	Procura Informada . . . . .	3
4.2.1	Gulosa . . . . .	3
4.2.2	A* (A estrela) . . . . .	3
<b>5</b>	<b>Resultados</b>	<b>3</b>
<b>6</b>	<b>Comentários finais e Conclusão</b>	<b>4</b>

# 1 Introdução

No âmbito da unidade curricular de Sistemas de Representação de Conhecimento e Raciocínio, foi-nos proposta a elaboração de um sistema capaz de importar dados relativos aos diferentes circuitos e representá-los numa base de conhecimento e, ainda, recomendar circuitos usando diferentes algoritmos.

Na minha resolução, optei por seguir a versão simplificada do enunciado.

## 2 Dataset

Numa fase inicial, comecei por decidir a estrutura da base de conhecimento, tendo em conta o *Dataset* fornecido.

Visto que optei pela versão simplificada do enunciado, decidi remover algumas colunas do *Dataset*, pois achei irrelevantes para a minha solução. Sendo assim, o *Dataset* usado contém as seguintes colunas: *Latitude*, *Longitude*, *PONTO\_RECOLHA\_LOCAL*, *CONTENTOR\_RESÍDUO* e *CONTENTOR\_TOTAL\_LITROS*.

Também tive que filtrar algumas linhas do *Dataset*, pois estas apenas diferenciavam no tipo de contentor. Sendo que essa coluna foi eliminada, iríamos ter linhas repetidas. Após a remoção destas linhas repetidas, o meu *Dataset* possui 137 entradas.

Após esta fase, procedi para o *parsing* do *Dataset*, através de um *script* em *python*.

### 2.1 parser.py

Na inicialização do programa, são criados dois dicionários: *datasetData* que guarda as informações de cada rua; *datasetDataByResidues* que guarda as informações de cada rua, agrupadas pelo tipo de lixo que estas contém.

Depois, também foram criados dois ficheiros de *output*. O ficheiro *pontos\_recolha.pl* guarda as informações relativos a todos os pontos de recolha e o ficheiro *arcos.pl* guarda as informações sobre cada arco.

De seguida, faço o *parsing* do conteúdo de cada linha do *Dataset*, guardando nas respetivas variáveis. Com estas variáveis fui capaz de popular os dicionários referidos anteriormente com a informação pertinente. Depois, as colunas de cada linhas são escritas para o ficheiro *pontos\_recolha.pl*, seguindo um dado formato.

Por fim, os arcos do grafo final são escritos para o ficheiro *arcos.pl*, tendo o cuidado de não repetir arcos.

## 3 Formulação do Problema

O problema exposto no enunciado do projeto consiste no estudo dos dados dos circuitos de recolha de resíduos urbanos do concelho de Lisboa. Neste problema em específico, os estados inicial e objetivo são ruas escolhidas por mim. O objetivo final é determinar os caminhos de recolha de resíduos começados e terminados nas ruas escolhidas, seguindo diferentes algoritmos de procura.

## 4 Estratégias de Procura

Nestas seção irei abordar as diferentes estratégias de procura usadas ao longo do trabalho.

### 4.1 Procura Não Informada

#### 4.1.1 Profundidade (DFS - Depth-First Search)

Este algoritmo caracteriza-se por obter o caminho mais comprido de um grafo. Tem uma complexidade temporal de  $O(b^d)$  e uma complexidade espacial de  $O(bm)$ .

Representando um dado grafo numa árvore,  $b$  é o fator da ramificação,  $d$  a profundidade da solução encontrada e  $m$  o tamanho da árvore.

No meu trabalho implementei este algoritmo tendo em conta um nodo inicial e final definidos previamente. Também implementei o algoritmo recebendo um tipo de lixo.

#### 4.1.2 Largura (BFS - Breadth-First Search)

Este algoritmo é usado quando queremos obter o caminho mais curto de um grafo. Tem uma complexidade temporal de  $O(b^d)$  e uma complexidade espacial de  $O(b^d)$ .

Neste caso não implementei o algoritmo tendo em conta os tipos de lixo, pois iria gastar muita memória e muito mais tempo.

#### 4.1.3 Busca Iterativa Limitada em Profundidade

Este algoritmo é muito parecido com o algoritmo DFS, sendo a única alteração o facto de receber um número, que indica o limite de nodos a ser pesquisados no grafo final.

Este algoritmo foi implementado tendo (e não tendo) em conta os tipos de lixo e tendo em conta os custos dos arcos (e consoante um tipo de lixo).

### 4.2 Procura Informada

#### 4.2.1 Gulosa

Este algoritmo aplica-se quando queremos obter um resultado, tendo em conta uma heurística. Neste caso escolhi a distância.

Carateriza-se por tomar decisões baseadas em estimativas. É um algoritmo rápido e simples, porém, as escolhas que tomas são definitivas, não permitindo voltar atrás e calcular outra solução.

#### 4.2.2 A\* (A estrela)

Este algoritmo é uma combinação do algoritmo de *Dijkstra* com uma BFS. É mais fidedigno que o algoritmo anterior.

Em cada nodo é aplicada a função heurística aos nodos vizinhos, sendo escolhido o nodo mais favorável. Depois, o processo é repetido até chegar ao nodo final.

## 5 Resultados

Para obter os resultados seguintes, escolhi como nodo inicial a 'R do Alecrim' e como nodo final a 'Av 24 de Julho'.

Estratégia	Tempo (segs)	Espaço	Profundidade/Custo	Melhor solução?
DFS	5	$O(bm)$	25	Sim
BFS	-	$O(b^d)$	-	Ind
BILP	-	-	-	Ind
Gulosa	0.001	-	14	Talvez
A*	0.001	-	14	Sim

Através da análise destes resultados, pode-se concluir que a BFS não é uma solução válida, devido à dimensão do grafo obtido. Em relação à BILP, não será utilizada pois o nosso objetivo é encontrar uma solução entre dois nodos especificados.

Em relação aos algoritmos de procura informada, estes são semelhantes em termos de tempo e custo. O algoritmo  $A^*$  é mais fidedigno que o algoritmo *Gulosa*.

## 6 Comentários finais e Conclusão

Terminada a implementação dos algoritmos e a análise dos resultados obtidos, posso concluir que, dentro dos algoritmos de procura não informada, aquele que se destaca mais é o DFS.

Em relação aos algoritmos de procura informada, o algoritmo  $A^*$  é o que produz resultados mais interessantes.

Posto isto, sinto que este trabalho ajudou a consolidar os temas abordados ao longo do semestre nesta unidade curricular, em particular no uso de grafos.

Concluindo, acho que o trabalho cumpre com os objetivos propostos no enunciado.