

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA**LICENCIATURA EM ENGENHARIA INFORMÁTICA****INTELIGÊNCIA ARTIFICIAL****2019/2020 – 2º semestre**

PROJETO – OTIMIZAÇÃO DA LOCALIZAÇÃO DE PRODUTOS NUM ARMAZÉM

1. Descrição do problema

O objetivo deste projeto é o desenvolvimento de uma aplicação que permita otimizar a localização dos produtos nas prateleiras de um armazém de modo a reduzir a distância necessária para a recolha desses mesmos produtos tendo em conta o historial de pedidos.

O programa deve começar por permitir que o utilizador escolha o problema a resolver. O problema é guardado num ficheiro de texto. A figura abaixo mostra um exemplo de um destes ficheiros.

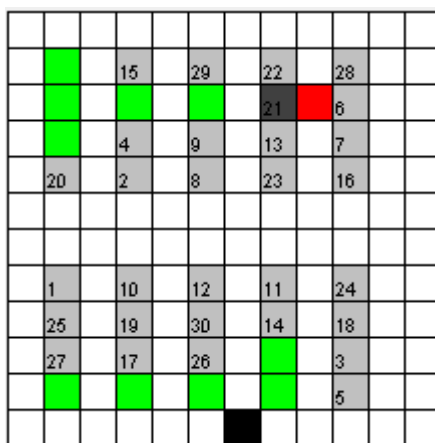
```
6
0 0 0 0 0 0
2 0 2 0 2 0
2 0 2 0 2 0
2 0 2 0 2 0
0 0 0 0 0 0
0 0 1 0 0 0
8
1 3 5 7 6 8
2 4 6 3 5 1
4 2 1
1 2 8
1 3 7 4
```

A primeira linha indica a largura/comprimento do armazém (vamos considerar que todos os armazéns são quadrados); Segue-se uma matriz que descreve a estrutura do armazém (uma célula a 0 indica um espaço livre; 1 representa uma porta; 2 representa uma prateleira); A linha seguinte indica o número de produtos cuja localização queremos otimizar; As restantes linhas representam o historial de pedidos dos produtos; por exemplo, a linha 2 4 6 3 5 1 indica um pedido em que o agente tem primeiro que recolher o produto 2, depois o produto 4, depois o produto 6, etc.

Considera-se que o agente que tem de recolher os produtos parte sempre da porta e, depois de recolher os produtos de um pedido, entrega-os também na porta. Em cada iteração, o agente pode mover-se para uma das células adjacentes Norte, Sul, Este, Oeste, desde que a célula exista

ou não esteja ocupada por uma prateleira. Considera-se que o agente recolhe um produto quando é a vez desse produto ser recolhido e o agente passa na célula adjacente à direita da célula onde o produto está localizado.

A figura abaixo mostra um armazém em que a localização dos produtos já foi determinada pelo algoritmo genético (células cinzentas com um valor numérico) e em que o agente (célula a vermelho) recolhe o produto 21. Depois de recolher todos os produtos de cada pedido, o agente recolhe à célula correspondente à porta (célula a preto).



2. Trabalho a realizar

O objetivo principal deste projeto é o desenvolvimento de uma aplicação que utilize o algoritmo de procura A* e algoritmos genéticos para resolver problemas como o descrito acima. Uma vez lido e processado o ficheiro de texto com o problema, sugere-se que este seja resolvido em três etapas:

- 1) Calcular as distâncias ótimas da porta a todas as prateleiras e as distâncias ótimas entre todos os pares de prateleiras. As distâncias devem ser calculadas recorrendo ao algoritmo A*.
- 2) Decidir qual a melhor localização para os produtos tendo em conta o historial de pedidos. A localização será tanto melhor quanto menor a distância total percorrida. Nesta etapa deve ser utilizado um algoritmo genético.
- 3) Para cada pedido, calcular o caminho a percorrer entre a porta (onde inicialmente se encontra o agente) e o primeiro produto a ser recolhido, entre cada dois produtos consecutivos solicitados no pedido e entre o último produto e a porta. Nesta etapa deve novamente ser utilizado o algoritmo A*.

No final, deve ser possível o utilizador visualizar a simulação da solução encontrada.

O projeto fornecido com este enunciado é baseado nos projetos desenvolvidos nas aulas e já contém alguns aspetos implementados, como sejam a leitura dos ficheiros que contêm os

problemas (datasets) e a GUI da aplicação, e outros parcialmente implementados, como sejam a representação do problema, o estado, o algoritmo A* e o algoritmo genético.

O projeto consiste na realização das tarefas seguintes:

1. Desenvolvimento da classe que permite representar um estado do problema (classe WarehouseState);
2. Desenvolvimento da classe que representa o problema, a ser utilizada pelo algoritmo A* (classe WarehouseProblemForSearch);
3. Desenvolvimento de uma heurística para o problema, a ser utilizada com o algoritmo A* (classe HeuristicWarehouse);
4. Desenvolvimento de classe que representa o problema, a ser utilizada pelo algoritmo genético (classe WarehouseProblemForGA);
5. Desenvolvimento da classe que permita representar um indivíduo do algoritmo genético (classe WarehouseIndividual);
6. Desenvolvimento de dois operadores de recombinação adequados ao problema e à representação utilizada para os indivíduos (classes Recombination2 e Recombination3);
Nota: já é disponibilizado um operador de recombinação (classe RecombinationPartialMapped).
7. Desenvolvimento de dois operadores de mutação adequados ao problema e à representação utilizada para os indivíduos (classes Mutation2 e Mutation3);
Nota: já é disponibilizado um operador de mutação (classe MutationInsert).
8. Realização de testes que permitam estudar os seguintes aspetos:
 - O efeito da variação do tamanho da população e do número de gerações;
 - Desempenho relativo dos diferentes operadores genéticos;
 - O efeito da variação das probabilidades dos operadores genéticos utilizados;
 - O efeito da variação do tamanho do torneio.

Do relatório do projeto deve constar:

- A descrição do estado;
- A descrição da heurística;
- A descrição da representação dos indivíduos utilizada no algoritmo genético;
- A descrição da função de avaliação (função de *fitness*) utilizada;
- A descrição do método de criação da população inicial;
- A descrição dos operadores genéticos desenvolvidos;
- A apresentação e discussão dos resultados obtidos nos testes realizados;
- Outros aspetos considerados relevantes para uma boa compreensão e avaliação do trabalho realizado.

Alguns dos fatores com mais importância na avaliação do relatório são:

- Clareza na descrição dos componentes da aplicação;
- A forma como os resultados dos testes são compilados e a clareza com que são apresentados (a utilização, mas não só, de tabelas e/ou gráficos pode ajudar);
- O significado estatístico dos resultados (que depende do número de *runs* realizados);
- A análise e discussão dos resultados.

3. Cotações

20% - Tarefa 1 e 2

5% - Tarefa 3

15% - Tarefa 4 e 5

20% - Tarefa 6 e 7

30% - Tarefa 8 e Relatório (devem ser entregues todos os ficheiros das experiências)

10% - Extras

Extras sugeridos:

- Considerar que cada produto terá n localizações diferentes no armazém. Algumas empresas (e.g. Amazon) utilizam este estratagema nos seus armazéns. O valor de n deve ser definido no ficheiro de texto do problema;
- Considerar que há x agentes em simultâneo a satisfazer pedidos. Neste caso, o algoritmo genético deverá não só tentar otimizar a distância percorrida pelos agentes mas também minimizar o número de colisões entre eles; Caso ocorram colisões, pode considerar que os agentes passam um pelo outro. O valor de x deve ser definido no ficheiro de texto.

4. Prazos, datas, regras e instruções

1. Data limite de entrega do projeto: **6 de junho de 2020**.
2. Data das defesas: **19 de junho de 2020**.
3. O projeto é realizado em grupos de 2 estudantes. Não são aceites projetos realizados por grupos com mais de 2 elementos. Os estudantes que pretendam realizar o projeto individualmente devem solicitá-lo, por escrito, ao docente responsável pela UC. Apenas em casos bem fundamentados serão autorizados projetos realizados individualmente.
4. O relatório deve ser realizado utilizando o modelo disponibilizado na secção *Projeto* do sítio da UC no Moodle.
5. O projeto deve ser entregue sob a forma de um arquivo zip, rar ou 7z que contenha todos os elementos do projeto, incluindo o relatório. O nome do arquivo deve ter o formato *IA_Projeto_#1_#2.(zip/rar/7z)*, onde #1 e #2 devem ser substituídos pelos números de estudante dos elementos do grupo. O relatório deve ser entregue em formato pdf e o seu nome deve ter a mesma estrutura do arquivo mas com extensão pdf.