

Exercício1) Pesquisa de Soluções (5 Valores)

Suponha o seguinte jogo (para 1 jogador) em que o tabuleiro é constituído por 9 casas e por peças com uma face branca e outra vermelha (tal como no Othello) que podem ser voltadas. O objetivo do jogo é que todas as peças fiquem brancas no tabuleiro. Para tal, o jogador pode, em cada jogada, trocar de cor um conjunto de peças que formem um quadrado (com 4 ou 9 peças). Suponha que cada jogada tem um custo correspondente ao número de peças voltadas (invertidas) na respetiva jogada. Suponha que o estado inicial é o seguinte:

V	B	V
V	B	V
B	B	B

1.1) Formule o problema como um problema de pesquisa, indicando o modo de representação do estado, estado inicial, teste objetivo, operadores (nome, pré-condições e efeitos) e função de custo.

Estado:

- Matriz M come 3 linhas e 3 colunas (elementos da matriz podem ter o valor 0 (branca) ou 1 (vermelha))

Estado inicial: $[[1,0,1], [1,0,1], [0,0,0]]$

Teste objetivo: Estado == $[[0,0,0], [0,0,0], [0,0,0]]$

Operadores (nenhum tem pré-condições):

- TL, Efeito: Inverte as cores das 4 peças mais acima e à esquerda. ($M[0][0] = 1 - M[0][0]$; $M[0][1] = 1 - M[0][1]$; $M[1][0] = 1 - M[1][0]$; $M[1][1] = 1 - M[1][1]$)

- TR, Efeito: Inverte as cores das peças mais acima e à direita

- BL, Efeito: Inverte as cores das peças mais abaixo e à esquerda.

- BR, Efeito: Inverte as cores das peças mais abaixo e à direita.

- ALL, Efeito: inverte as cores de todas as peças da matriz.

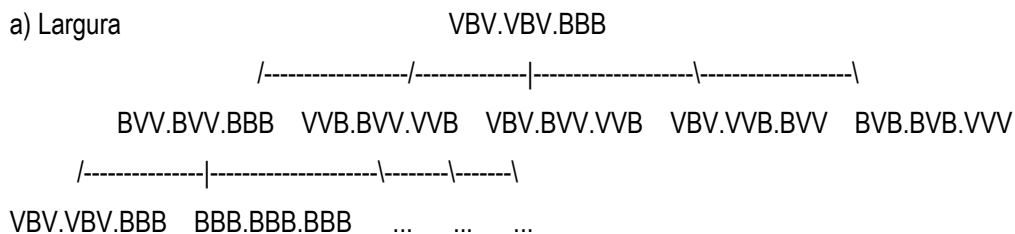
Apenas o operador ALL tem custo 9, os outros têm custo 4. A função de custo é o número total de peças invertidas.

1.2) Qual o fator de ramificação médio da árvore de pesquisa? Qual a dimensão do espaço de estados do problema? Justifique detalhadamente.

Cada nó da árvore de pesquisa dá origem a 5 sucessores, pois há 5 operadores possíveis (todos sem pré-condições), logo, o fator de ramificação médio é 5.

Cada elemento da matriz pode ter 2 valores e a matriz tem 9 elementos. Logo, a dimensão do espaço de estados (número de estados possíveis para o problema) será $2^9 = 512$.

1.3) Supondo o estado inicial apresentado, aplique os métodos **Pesquisa em Largura** e **Pesquisa em Profundidade Limitada** (Limite=4) para resolver este problema. Apresente as respectivas árvores de pesquisa geradas. Utilize os símbolos / | \ para ligar os nós da árvore. Considere que, em ambos os métodos, quando um nó é expandido são gerados imediatamente todos os seus sucessores e que a ordem de geração corresponde a gerar primeiro os quadrados de 4 peças (começando em cima e em caso de empate à esquerda) e só no final o quadrado de 9 peças. Considere que não é utilizado qualquer método para evitar estados repetidos.



b) Profundidade

Exatamente Igual ao anterior

1.4) Defina uma função admissível **h(n)** para aplicação do algoritmo **A***, explicando porque é admissível e apresentando o respetivo pseudo-código.

$h(n)$ = número de peças vermelhas no tabuleiro (admissível porque nunca sobrestima o custo para chegar à solução pois precisamos pelo menos de inverter as peças vermelhas que estão no tabuleiro)

```

function h(state)
  cnt = 0;
  for row = 1 to 3
    for col = 1 to 3
      if ((M[row,col] == 1) cnt++;
  return cnt;
  
```

Exercício 2) Optimização (5 Valores)

Uma entidade governamental com competências na área a inspeção alimentar necessita de realizar, periodicamente, inspeções a diversos estabelecimentos. Para o efeito, tem ao seu dispor duas brigadas de inspeção, para as quais gera, diariamente, rotas de inspeção. A execução dessas rotas deve caber num dia de trabalho (8h), pelo que na geração das rotas há que ter em conta o percurso de viagem e o tempo despendido em cada inspeção, no local do estabelecimento em causa. Dado um conjunto de n estabelecimentos, cada um devidamente georreferenciado (por simplificação considere as coordenadas (x,y) em km), pretende-se gerar duas rotas de inspeção, procurando maximizar o número de estabelecimentos inspecionados. Como simplificação suponha que para cada estabelecimento existe um tempo de inspeção de uma hora e que o tempo de deslocação, em horas, de um ponto para outro é igual a 1/20 da distância Manhattan entre esses pontos (por exemplo, entre os pontos (3,3) e (8,8) o tempo de deslocação será 0.5 horas (meia hora).

2.1) Sugira uma representação para os ficheiros contendo instâncias a resolver do problema. Sugira também uma representação para a solução deste problema, onde devem ser considerados os estabelecimentos, ordenados, a visitar por cada brigada, tendo em conta a existência de diversos estabelecimentos que não serão inspecionados.

Para a representação em ficheiro, assumindo que o fator de $1/20$, o tempo total do dia de trabalho e o número de brigadas é constante, cada instância teria o número de estabelecimentos na primeira linha, seguido das suas coordenadas no formato "x y". Exemplo:

```
5
1 10
5 4
20 2
12 4
2 8
```

A solução consistiria em três listas de inteiros (assumindo duas brigadas), tendo as duas primeiras os índices dos estabelecimentos que cada brigada visita, por ordem e a terceira os índices dos estabelecimentos que não serão visitados.

2.2) Identifique a restrição forte e o critério de otimização do problema. Explique devidamente como é interpretada e avaliada a solução, exemplificando e apresentando um pseudo-código para o cálculo da respetiva função de avaliação.

Como restrição forte temos que a soma total de tempos de inspeção e deslocação não pode exceder 8h para cada brigada. Como critério de otimização, temos de maximizar o número de estabelecimentos visitados.

Para a função de avaliação, basta contar o número de estabelecimentos (não repetidos) visitados, penalizando as soluções que excedam o tempo limite.

```
function evaluate(solution)
```

```
    return lista1.size() + lista2.size()
```

(resposta poderia ser mais completa contando o tempo e penalizando soluções inválidas)

2.3) Considere os seguintes estabelecimentos e suas localizações num mapa em grelha: A (20;20), B (10;10), C (10;0), D (10;20), E (20;10). Considerando que as brigadas partem do ponto (0;0), ao qual têm que voltar no final da jornada, e considerando que o tempo de deslocação, em horas, é igual a $1/20$ da distância Manhattan, determine se a solução na qual a brigada 1 visita A, B e C, e a brigada 2 visita D e E é viável. Justifique, indicando quantas horas seriam necessárias para cada brigada.

Distância percorrida por 1: $20 + 20 + 10 + 10 + 10 + 10 = 80$

Tempo de deslocação: $80 / 20 = 4h$

Tempo de inspeção: $1 * 3 = 3h$

Tempo total: $4 + 3 = 7h$

Distância percorrida por 2: $10 + 20 + 10 + 10 + 20 + 10 = 80$

Tempo de deslocação: $80 / 20 = 4h$

Tempo de inspeção: $1 * 2 = 2h$

Tempo total: $4 + 2 = 6h$

A solução é viável, pois o tempo total de cada brigada não excede 8h (e é ótima pois visita todos os estabelecimentos).

2.4) Com vista a aplicar um algoritmo de pesquisa local (por exemplo, arrefecimento simulado), defina uma função de vizinhança. Exemplifique a sua utilização para o exemplo da alínea anterior, identificando duas soluções vizinhas diferentes, de acordo com a representação que definiu.

Uma função de vizinhança simples poderia ser:

1. Escolher um número aleatório entre 0 e 2, para escolher se vamos remover um elemento, adicionar um elemento ou trocar um elemento entre brigadas

2.1 Se remover, escolher uma brigada aleatória e um elemento aleatório da brigada a remover, apagar esse elemento e adiciona-lo na lista de não visitados

2.2 Se adicionar, escolher um elemento aleatório da lista de não visitados, apagar o elemento, escolher uma brigada a adicionar e uma posição aleatória para o colocar e coloca-lo na posição escolhida

2.3 Se trocar, escolher um elemento aleatório de cada brigada e troca-los

Por exemplo:

Solução: [ABC, DE, []]

Possíveis vizinhos (2.1): [BC, DE, A], [AC, DE, B], [AB, DE, C], [ABC, E, D], [ABC, D, E]

Possíveis vizinhos (2.3): [DBC, AE, []], [EBC, DA, []], [ADC, BE, []], [AEC, DB, []], [ABD, CE, []], [ABE, DC, []],

Nota: Existem muitas outras funções de vizinhança possíveis mas têm de permitir explorar todo o espaço de estados. Por exemplo, uma função que simplesmente troque um elemento aleatório entre brigadas não chega pois não permite adicionar e remover elementos nos percursos das brigadas.

GRUPO3: Inteligência Artificial (10 valores)

Resolva Exclusivamente 6 das seguintes 9 questões

9. Apresente a descrição PEAS de um agente para o Piloto Automático de um Avião que realiza exclusivamente as aterrisagens de um avião comercial, descrevendo detalhadamente as características do ambiente. Acha possível usar um agente reativo simples para controlar um agente deste tipo? Justifique.

Medida de desempenho: Aterragem segura, não brusca, confortável para os passageiros

Ambiente: Via aérea (pássaros, outros aviões) e pista de aterragem (pessoas, veículos, outros aviões)

Atuadores: Mecanismos de controlo do avião, controlo do rol e pitch, controlo da potência dos motores, controlo do trem de aterragem, etc.

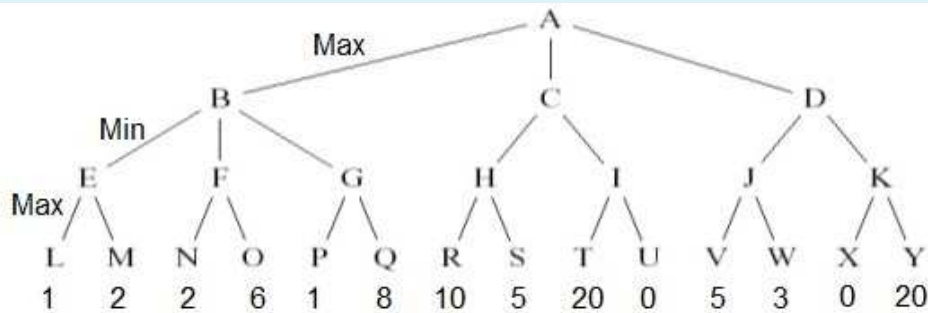
Sensores: Sensor de altura, inclinação, pressão, velocímetro, GPS, giroscópio, etc.

O ambiente é dinâmico, inacessível, não determinístico e contínuo.

O mundo real é bastante complexo para um agente reativo simples, especialmente quando envolve uma tarefa complexa e crítica (vida dos passageiros) que não pode ser resolvida simplesmente com ligação direta entre o valor dos sensores para as ações. Por isso não é possível usar um agente reativo simples.

10. Aplique o algoritmo minimax com cortes alfa-beta à seguinte árvore, supondo que joga primeiro MAX, depois MIN e novamente MAX, indicando:

- a) Qual o valor final dos Nós A, B, C e D? (Nota: dado que são usados cortes alfa-beta o valor pode ser um número, ex: 8 ou uma condição ex: ≥ 20 ou ≤ 8)
- b) Quais os nós folha (de entre L a Y) que não chegam a ser avaliados pelo algoritmo minimax com cortes alfa-beta?



- a) A = 10, B = 2, C = 10, D ≤ 5
- b) O, U, X, Y

11. Comente a seguinte afirmação justificando: "A aplicação do algoritmo minimax em jogos complexos permite obter decisões rápidas mesmo que imperfeitas e gerir qual o tempo máximo para obter a solução. Por exemplo, podemos definir um tempo máximo de 30 segundos para obter uma solução e o algoritmo pode retornar uma solução ao fim desses 30 segundos".

O algoritmo minimax (usando cortes alfa-beta) e com profundidade limitada pode ser utilizado para jogos complexos. Como o algoritmo analisa as jogadas possíveis, primeiro em profundidade, encontra uma primeira solução muito rápida (ao chegar à profundidade limite). Para retornar uma solução ao fim de um tempo limite basta interromper a pesquisa e verificar qual a melhor solução até esse momento retornando-a. Claro que esta solução rápida será imperfeita o que pode ser melhorado com uma boa ordenação dos nós na expansão (colocando primeiro as jogadas mais prometedoras).

12. Apresente o pseudo-código necessário para a aplicação de arrefecimento simulado na resolução do problema 2 deste exame, utilizando um decréscimo de temperatura linear (de 1.0 até 0.0) entre as iterações 0 e 1000 e realizando um total de 1500 iterações (as últimas 500 a temperatura 0). Não necessita de copiar as funções que definiu nas alíneas da pergunta 2 (ou se não as definiu basta utiliza-las neste exercício como se estivessem definidas na pergunta 2).

```

current = initial_solution()
for it = 1 to 1500
    if (it > 1000) T = 0 else T = 1.0 - it/1000
    next = randomNeighbor(current);
    diff = evaluate(next) - evaluate(current);
    if (diff < 0 || random(0, 1) < e ^ (diff / T)) current = next
  
```

13. O Mundo de Wumpus pode ser resolvido por um agente puramente reativo? Justifique.

Não! O ambiente do mundo de Wumpus não é totalmente acessível e precisa de ser explorado. É preciso que o agente se lembre de estados anteriores, onde é que o "stench" e "breeze" estiveram, etc. Por isso, um agente reativo

simples, embora, com sorte possa resolver algumas instâncias do problema, não pode com segurança resolver o mundo do wumpus.

14. Suponha os seguintes conjuntos de treino (indivíduos 1 a 13) e de teste (indivíduos 14, 15 e 16). Sem necessidade de fazer contas, supondo a distância Euclidiana, indique, justificando, quais as classes previstas para os indivíduos 14, 15 e 16 utilizando o algoritmo Nearest Neighbour, versões 1NN e 3NN.

1NN: 1 vizinho mais próximo

14: C ; 15: B ; 16: A

3NN: 3 vizinhos mais próximos

14: B ; 15: B ; 16: A



Treino	Par1	Par2	Par3	Classe
Ind1	3	8	9	A
Ind2	4	7	8	A
Ind3	2	2	5	B
Ind4	1	3	6	B
Ind5	8	6	5	C
Ind6	9	5	6	C
Ind7	2	1	6	B
Ind8	1	3	5	B
Ind9	4	9	8	A
Ind10	3	8	7	A
Ind11	9	4	6	C
Ind12	8	5	5	C
Ind13	1	1	1	C

Teste	Par1	Par2	Par3	Classe 1NN	Classe 3NN
Ind14	1	1	3	?	?
Ind15	2	4	6	?	?
Ind16	5	9	9	?	?

15. Suponha que dispõe de um ficheiro CSV com dados relativos à dimensão (em m2), localização (concelho), ano de construção (valor inteiro) e tipo de acabamentos (escala com 5 valores possíveis) de 20000 imóveis portugueses. Para cada imóvel está ainda disponível no CSV uma classificação do seu "tipo para venda"(simples, médio, luxo e super_luxo). Suponha que pretende usar aprendizagem supervisionada para treinar um classificador para, a partir da dimensão, localização, ano e tipo de acabamentos ser capaz de classificar o "tipo para venda" de novos imóveis. Apresente o código essencial (resumido) para, utilizando Python e as bibliotecas Pandas e SciKit Learn, ler o CSV, treinar um classificador com Nearest Neighbour ou SVMs utilizando 10 fold cross validation e apresentar os resultados de três das seguintes medidas: accuracy, precision, sensitivity e/ou f-measure. Nota: Não necessita de apresentar o código absolutamente correto bastando uma aproximação.

```
import pandas as pd
import sklearn as sk
data=pd.read_csv("ficheiro.csv")
treino=['direcao','localizaca','ano']
dados_treino=data[treino]
dados_teste=data['tipo']
x_treino,x_teste,y_treino,y_teste=sk.split(dados_treino,dados_teste,0,2)
modelo=sk.KNearestneighbour()
modelo.fit(x_treino,y_treino)
model.acuracy(x_teste,y_teste)
model.sensitivitu(x_teste,y_teste)
model.precision(x_teste,y_teste)
... (solução feita sem consultar documentação mas era esse o objetivo).
```

16. Num problema de aprendizagem por reforço, ao chegar a um estado um agente tem ao seu dispor quatro ações, cujos valores Q são os seguintes: $Q(A)=5$, $Q(B)=5$, $Q(C)=1$, $Q(D)=6$. Seguindo uma política de seleção de ações do tipo soft-max, qual é a probabilidade, arredondada a 2 casas decimais, de escolha da ação B quando a temperatura é de 0.5?

$$P = (e^{5/0.5}) / (e^{5/0.5} + e^{5/0.5} + e^{1/0.5} + e^{6/0.5}) = 0.11$$

17. Considere o texto desta pergunta, que tem um vocabulário de tamanho 27 (número de sequências diferentes de caracteres alfanuméricos). Qual é a probabilidade de ocorrência da palavra "de", usando Laplace smoothing?

$$P = (\text{count}(\text{"de"}, c) + 1) / (\text{count}(w, c) + \text{voc_size}) = (5 + 1) / (27 + 31) = 0.10$$