Respostas questionário – Tarefa 5
Tomás Abril

1)

agente.py
função busca_lrta

A heurística utilizada foi a euclidiana pois leva em conta a possibilidade de andar na diagonal. Ela continua admissível porque o menor custo possível para um passo é um. E na heurística o valor máximo para cada passo é 1, sem falar que existem paredes no caminho.

2)

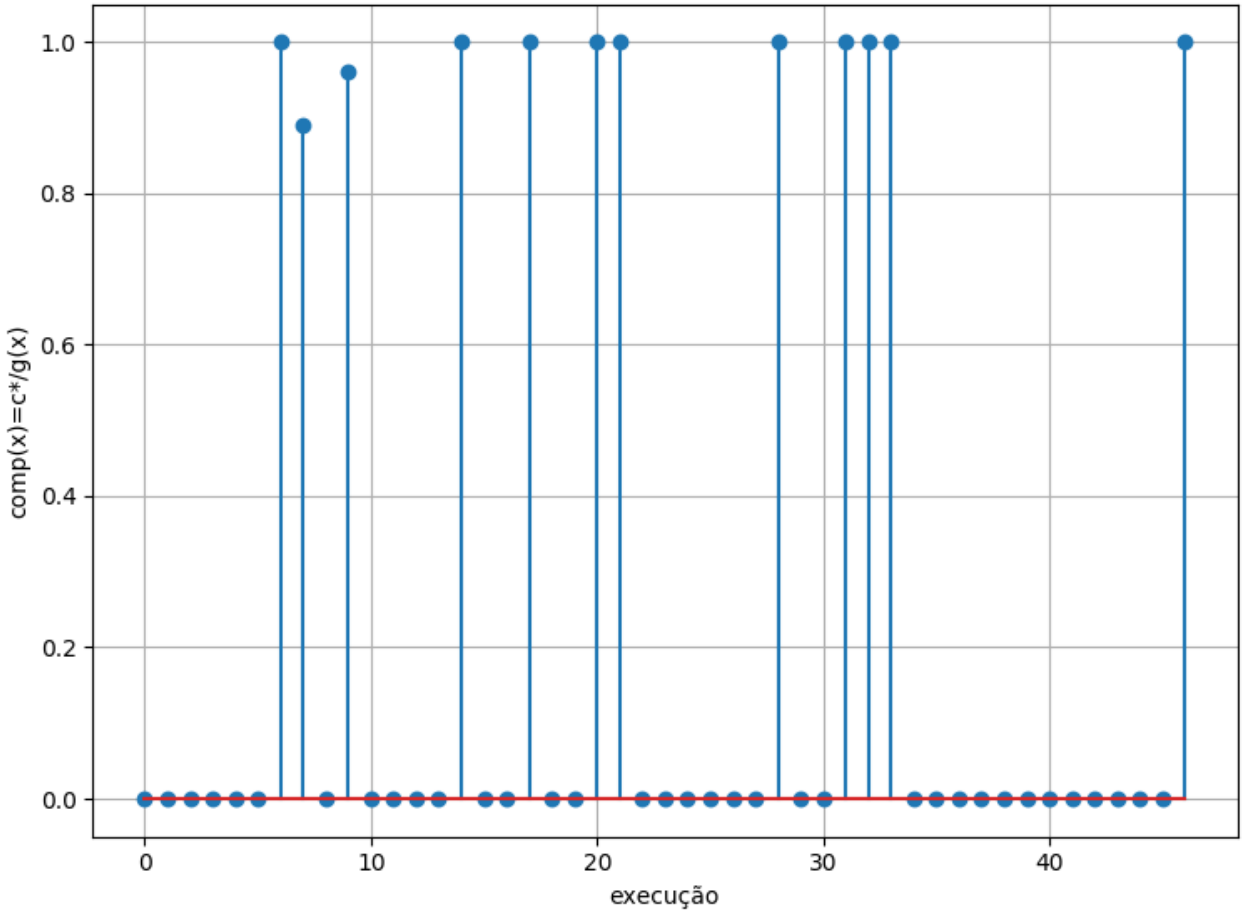
custo minimo: 12.0
7 caminhos ótimos
[8, 8, 9, 9, 8, 8, 9, 6, 6, 3]
[8, 8, 9, 9, 6, 6, 6, 6, 9, 7]
[8, 8, 9, 8, 8, 9, 9, 6, 6, 3]
[8, 8, 8, 8, 9, 9, 9, 6, 6, 3]
[8, 8, 9, 8, 9, 8, 9, 6, 6, 3]
[8, 8, 8, 9, 9, 8, 9, 6, 6, 3]
[8, 8, 8, 9, 8, 9, 9, 6, 6, 3]

legenda (como no teclado numérico):
7 ↖
8 ↑
9 ↗
6 →
3 ↘
2 ↓
1 ↙
4 ←

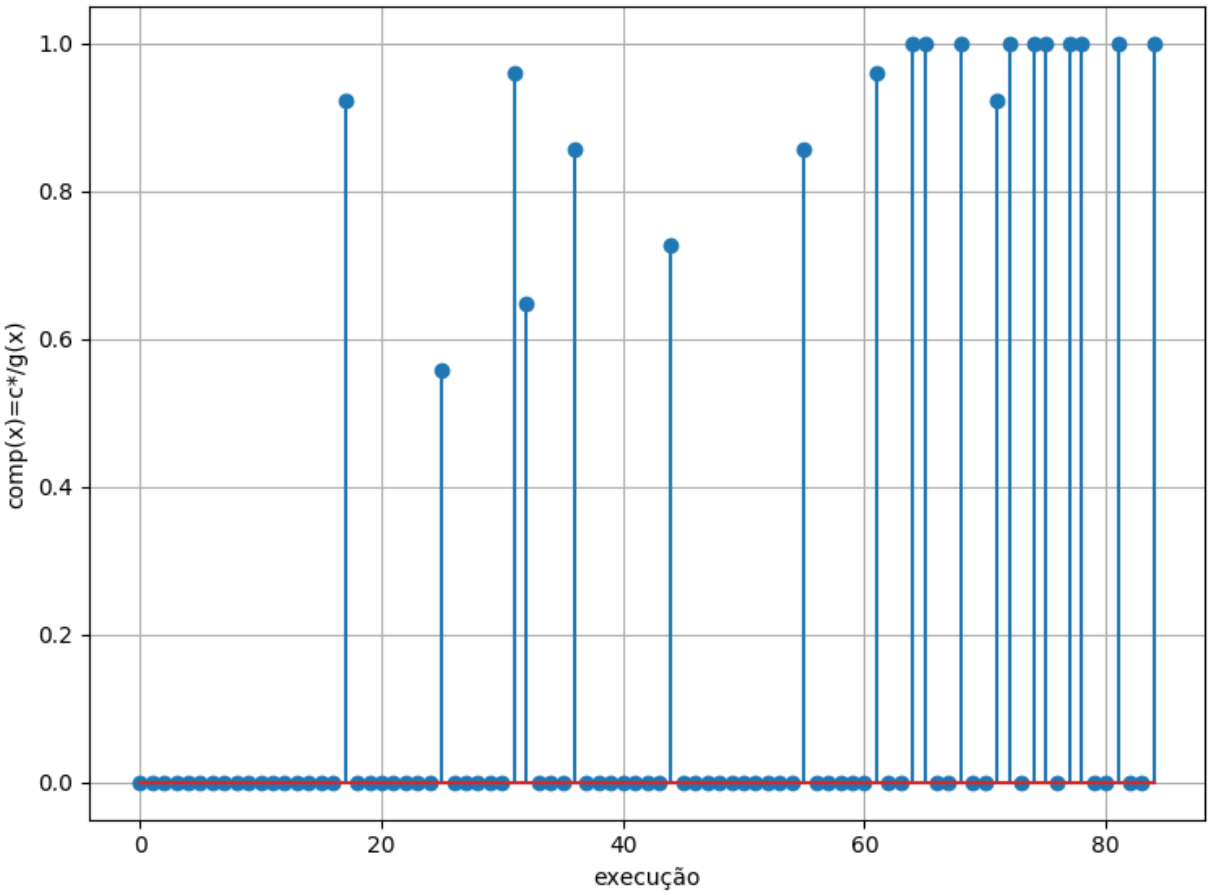
3)

Comendo todas as frutas o agente é capaz de chegar ao destino em aproximadamente 17,5% das vezes. Comendo 50% das frutas em aproximadamente 1,1% das vezes.

4)



5)



6)

@relation energia-da-fruta

@attribute madureza {verde, madura, podre}
@attribute carboidratos {pouca, moderada, alta}
@attribute fibras {pouca, moderada, alta}
@attribute proteinas {pouca, moderada, alta}
@attribute lipideos {pouca, moderada, alta}
@attribute delta_energia {-10, 100, 160}

@data

7)

O arquivo utilizado tem 653577instâncias.
24,7 MB

8)

Todos foram utilizados.

9)

Para o teste foram criados dois arquivos .arff, ambos com aproximadamente seiscentos e cinquenta mil instâncias. Um deles foi utilizado como treino (Use training set) e o outro para o teste (Supplied test set).

10)

A árvore de decisão gerada foi:

```
madureza = verde
|  lipideos = pouca
|  |  carboidratos = pouca
|  |  |  proteinas = pouca: -10
|  |  |  proteinas = moderada: -10
|  |  |  proteinas = alta
|  |  |  |  fibras = pouca: -10
|  |  |  |  fibras = moderada: -10
|  |  |  |  fibras = alta: 100
|  |  carboidratos = moderada: -10
|  |  carboidratos = alta: -10
|  lipideos = moderada
|  |  carboidratos = pouca: -10
|  |  carboidratos = moderada: 100
|  |  carboidratos = alta: 100
|  lipideos = alta
|  |  carboidratos = pouca: -10
|  |  carboidratos = moderada: 100
|  |  carboidratos = alta: 100
madureza = madura
|  carboidratos = pouca
|  |  lipideos = pouca
|  |  |  fibras = pouca: -10
|  |  |  fibras = moderada: -10
|  |  |  fibras = alta
|  |  |  |  proteinas = pouca: -10
|  |  |  |  proteinas = moderada: -10
|  |  |  |  proteinas = alta: 100
|  |  lipideos = moderada: 100
|  |  lipideos = alta: 100
|  carboidratos = moderada: 160
|  carboidratos = alta: 160
madureza = podre: -10
```

A implementação dela se encontra no arquivo agente.py no método id3table_final

11)

Como pode ser visto abaixo todos as instâncias de teste passaram.
F-mesure e recall são 1 porque nenhum teste foi classificado erroneamente.
Na matriz de confusão também está claro que nenhum teste foi classificado em um grupo diferente do seu.

=== Summary ===

Correctly Classified Instances	653354	100 %
Incorrectly Classified Instances	0	0 %
Kappa statistic	1	
Mean absolute error	0	
Root mean squared error	0	
Relative absolute error	0 %	
Root relative squared error	0 %	
Total Number of Instances	653354	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	-10
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	100
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	160
Weighted Avg.	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	

=== Confusion Matrix ===

a	b	c	<-- classified as
357658	0	0	a = -10
0	150493	0	b = 100
0	0	145203	c = 160

12)

agente.py método ver_fruta

if fruta ganha mais do que gasta pra comer (de acordo com arvore do id3):
– if minha energia > 440:
– – guardar fruta
– else:
– – comer fruta
if tem fruta guardada and energia < 440:
– comer fruta guardada

13)

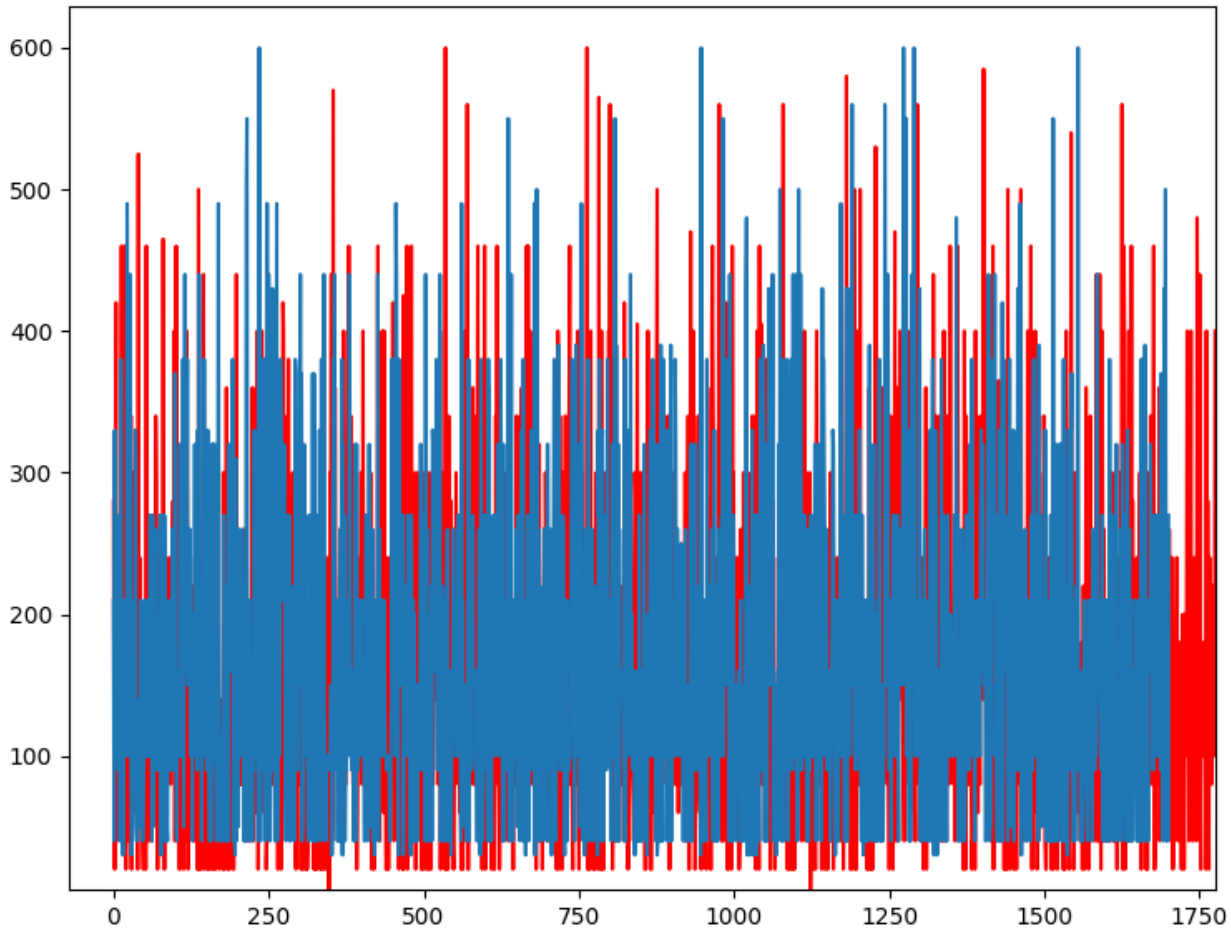
O agente chega até o fim em aproximadamente 25% das execuções.

14)

Pela imagem é difícil ver, mas comparando as médias pode-se perceber que com o algoritmo id3 o agente tem menos energia no final em comparação com comer todas as frutas e mais em comparação com comer metade das frutas.
Um fato a se notar é a quantidade de vezes que o agente chega no fim com cada possibilidade.
1746/10000 vezes comendo todas as frutas
0105/10000 vezes comendo 50% das frutas.
2519/10000 vezes com regras do ID3.

Media de energia no fim comendo todas as frutas: 166,167
Media de energia no fim comendo 50% das frutas aleatoriamente: 94.381
Media de energia no fim com id3: 156,881

Na figura: em azul a energia restante ao chegar no objetivo comendo todas as frutas, em vermelho a energia restante comendo as frutas de acordo com a árvore do id3.



15)

Medida de desempenho com id3 = 2,21527440405651
Medida de desempenho comendo 100% = 1.6969435491416165
Medida de desempenho comendo 50% = 1,222606231772832

(medida de desempenho) = (media de energia restante)/(máxima energia restante)
+ (custo mínimo)/(media de custos)
+ (vezes que chegou ao fim)/(vezes que chegou ao fim comendo aleatoriamente + vezes com id3)

A ideia dessa medida de desempenho é que cada característica some um valor positivo ao valor final. Quanto mais energia restante maior o valor somado. Quanto menos a media dos custos do caminho maior o valor somado. Quanto mais vezes chegou no fim maior o valor somado.
As divisões existem para equalizar os valores e deixa-los com pesos mais parecidos.

A implementação dela se encontra no arquivo main.py