Faculadde de Ciências e Tecnologia

Inteligência Artificial

Relatório do 1º Trabalho Prático

Autores:
Daniel Parreira 28436
João Ferreira 28543
Laura Oliveira 28728

Professor:
Carlos Viegas Damásio

Grupo 11
Turno Prático 4

24 de Outubro de 2010

Relatório do 1º Trabalho Prático

Daniel Parreira nº28436
João Ferreira nº28543
Laura Oliveira nº28728

24 de Outubro de 2010

Resumo

Este projecto tem como objectivo a implementação de um algoritmo de procura, AStar (A*), e a comparação deste com outros algoritmos. Para isso vão ser usados três problemas de procura: VacumWorld, NPuzzle e MarsRover. Este último foi implementado pelos alunos, assim como a heurística a utilizar, consistindo este último ponto o verdadeiro desafio : encontrar a melhor heurística possível. Vão ser apresentados resultados dos testes efectuados, bem como uma análise às várias heurísticas pensadas.

1 VacumWorld

1.1 Resultados

Estado Inicial: {Posição 1}
Mundo: {true, true, false, false, true}

Algoritmo	Nós Expandidos	Nós Gerados	Custo
DepthFirst	12	36	11
UniformCost	29	88	8
BreadthFirst	29	88	8

Estado Inicial: {Posição 1}
Mundo: {true, true, false false, true, false, false, true}

Algoritmo	Nós Expandidos	Nós Gerados	Custo
DepthFirst	26	78	21
UniformCost	91	274	12
BreadthFirst	87	262	12

2 nPuzzle

2.1 Resultados

Estado Inicial: { {7,2,4}, {5,null,6}, {8,3,1} }
Estado Final (Objectivo): { {1,2,3}, {4,null,5}, {6,7,8} }

Algoritmo	Nós Expandidos	Nós Gerados	Custo
UniformCost	141768	379535	24
BreadthFirst	140521	376039	24
*A**	442	1260	24

Estado Inicial: { {1,3,5}, {4,null,2}, {6,7,8} }

Estado Final (Objectivo): { {1,2,3}, {4,null,5}, {6,7,8} }

Algoritmo	Nós Expandidos	Nós Gerados	Custo
UniformCost	32	89	4
BreadthFirst	24	65	4
*A**	4	13	4

Estado Inicial: { {7,2,4,9}, {5,null,6,10}, {8,3,1,11}, {12,13,14,15} }
Estado Final (Objectivo): { {1,2,3,4}, {5,null,6,7}, {8,9,10,11}, {12,13,14,15} }

Algoritmo	Nós Expandidos	Nós Gerados	Custo
UniformCost	Out of memory	Out of memory	Out of memory
BreadthFirst	Out of memory	Out of memory	Out of memory
*A**	35056	115727	32

3 MarsRover

3.1 Representação de um estado

- Dois inteiros para a localização do rover no mapa
- BitmapTerrain para representar o mapa sobre o qual o MarsRover se desloca

3.2 Operadores

Os operadores que consideramos para a modelação deste problema foram os 8 pontos cardiais pelos quais o MarsRover se pode movimentar (North, West, East, South, Southeast, Southwest, Northwest e Northeast).

3.3 Custo do caminho

O custo do caminho é dado pela distância percorrida pelo MarsRover. Este custo pode ser multiplicado em função do tipo de terreno (Plain, Sand e Rock) sobre o qual se desloca. Além disso existe um factor multiplicativo que mitiga/potencia o custo do deslocamento caso este seja a descer ou a subir, respectivamente.

3.4 Heurísticas - Análise por ordem de concepção

3.4.1 Distância Euclidiana pelo eixo *xy* (sem altura)

Assumindo que a Distância Diagonal é admisivel, e como a distância entre dois pontos no plano é sempre inferior ou igual à distância diagonal entre os pontos, admitimos que esta heurística também ela, é admissivel.

3.4.2 Distância Euclidiana pelo eixo *xyz* (com altura)

Não é admisivel porque basta pensar no cenário com uma descida ligeira para a heurística falhar. A sua fraquesa é o factor multiplicativo das descidas, que faz com que a heurística devolva um resultado superior à custo total para o alvo, impossibilitando que seja admissivel. Numa descida, a distância entre o inicio e o fim da descida é dada pela distância euclidiana, no entanto, devido às regras do problema, na realidade o custo é essa distância a multiplicar por 0.98, o que torna o valor mais pequeno. Como a heurística para este caso devolve um custo estimado superior ao custo real, a heurística não é admissivel.

3.4.3 Distância Euclidiana pelo eixo *xyz* (com altura) e a compensar as descidas para a tornar admissivel

Atendendo à heurística anterior podemos perceber que a sua "fraqueza"é o facto de nas descidas o custo real ser inferior ao estimado. Para remediar isto, embora à custa de velocidade da heurística, sempre existe um diferença entre as altitudes de dois pontos, consideramos o "pior"caso, ou seja, de a descida ser feita o mais gradualmente possivel (pois assim o impacto do 0.98 é maior, já que 0.98 elevado um número maior que 1 diminui-se cada vez mais gradualmente), fazendo assim com que a heurística nunca estime valores acima do custo real. No entanto, esta correcção pode em certos cenários provocar um efeito contraproducente na heurística que faz com que o Rover explore primeiro o topo das montanhas.

3.4.4 Distância Diagonal pelo eixo xy (sem altura) (a escolhida por ser admissível)

Sem altura, esta heurística mostra-se admissivel. Se a altura entre os dois ponto for igual, a heurística estima o custo real na perfeição. Se existir subidas ou descidas, a heurística continua a estimar abaixo do custo real, não quebrando assim a sua admissibilidade. Quando existe uma subida, além de a distância pura entre os dois pontos só por si aumenta, que por sua vez é potenciado pelo factor multiplicativo do custo numa subida de 1.02. Nas descidas, o raciocinio não é tão simples. Será possivel existir uma descida tão vertiginosa que faça o factor multiplicativo da descidas de 0.98 anular a diferença do acrescimo da distância pura entre os dois pontos de modo a que essa descida se torne seja menos dispendiosa do que andar a direito? A resposta é: não. E segue-se a demonstração num cenário de uma descida. A altura da descida e o Δ do deslocamento horizontal são as variáveis do problema.

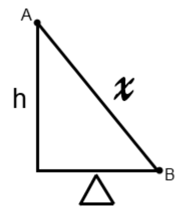

Figura 1: Distância real (visto pelo eixo xz)

Figura 2: Distância Diagonal (visto pelo eixo xy)

Logo o nosso objectivo é provar que

$$\Delta \leq x \tag{1}$$

se verifica para todos os valores possiveis de h (e os dois possiveis de Δ, que é 1 ou $\sqrt{2}$ quando se desloca na diagonal)

$$x^2 = h^2 + \Delta^2 \iff x = \sqrt{h^2 + \Delta^2}$$

$$\Delta \leq x \iff \Delta \leq \sqrt{h^2 + \Delta^2}$$

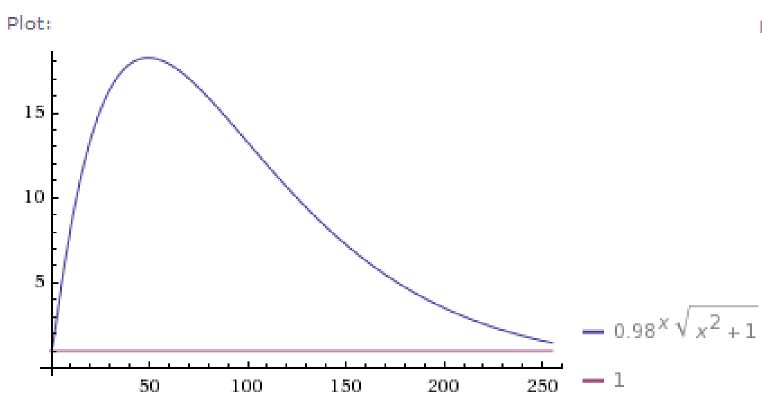

Figura 3: Gráfico para $\Delta = 1$

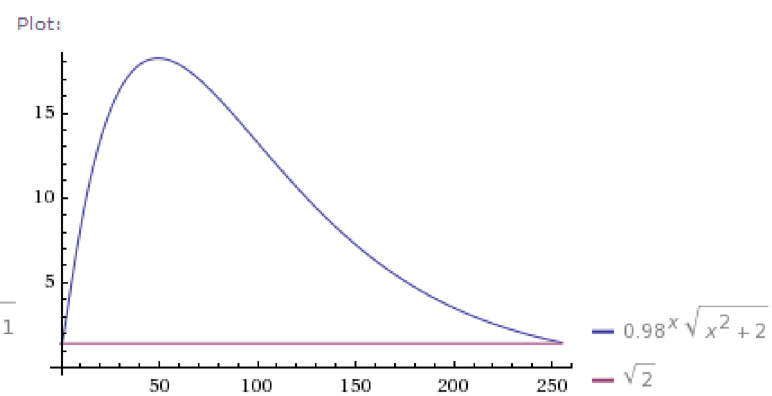

Figura 4: Gráfico para $\Delta = \sqrt{2}$

Para se observar melhor, os gráficos da diferença do custo real e a heurística.

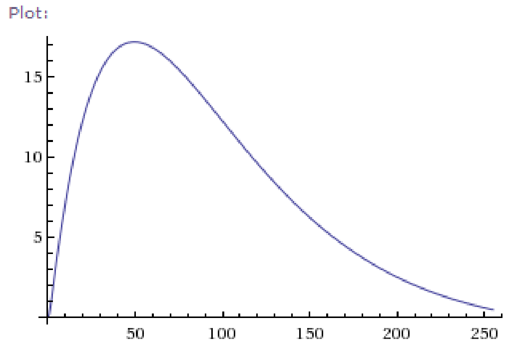

Figura 5: Gráfico para $\Delta - 1$

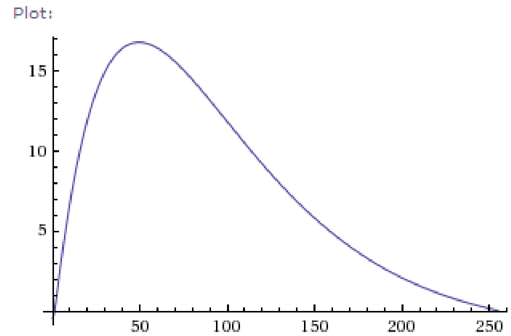

Figura 6: Gráfico para $\Delta - \sqrt{2}$

Recorrendo a um *graphical plotter* como o *WolframAlpha*, facilmente se observa que o valor do Δ é sempre inferior ou igual ao valor real percorrido (o x), ou seja, não existe nenhum x grande o suficiente (dentro do nosso universo, em que a altura varia entre 0 e 255) que faça a heurística devolver um valor superior ao valor real do custo, provando assim que é **admissivel**.

3.4.5 Distância Diagonal pelo eixo *xyz* (com altura)

Devido à complexidade da heurística, é nos impossivel demonstrar que esta heurística é admissivel. Para isso seria necessário resolver vários sub-problemas como por exemplo, se existe transitividade entre o custo de um caminho face apenas às suas coordenadas e altitude iniciais e finais. Depois de provado, seria possivel demonstrar se a heurística é admissivel ou não atráves da análise dos vários casos particulares de subida e descida. Nenhum dos testes efectuados apontam para que não seja admissivel (comparando com outras heurísticas admissiveis, que por sua vez foram comparadas com o custo obtido pelo UniformCostSearch). O problema com a demonstração desta heurística reside no facto de, ao contrário da compensação para descidas na Distância Euclidiana, usarmos também o factor multiplicativo das subidas numa tentativa de aproximar o máximo possivel a função heurística do custo real, de forma a minimizar o número de nós explorados. Se fosse omitido este último factor, esta heurística ficaria admissivel, da mesma forma que a Distância Euclidiana adaptada.

3.5 Resultados das Heurísticas

Estado Inicial: X = 141 ; Y = 512
Estado Final (Objectivo): X = 180 ; Y = 540

Heurística	Nós Expandidos	Nós Gerados	Custo
Euclidiana em xy	12124	96993	174.1485009717627
Euclidiana em xyz	9673	77385	174.1485009717627
Euclidiana em xyz *compensada*	12053	96425	174.1485009717627
Diagonal em xy	11597	92777	174.1485009717627
Diagonal em xyz	9061	72489	174.1485009717627

Estado Inicial: X = 141 ; Y = 512
Estado Final (Objectivo): X = 400 ; Y = 512

Heurística	Nós Expandidos	Nós Gerados	Custo
Euclidiana em xy	55031	440249	475.211216084949417
Euclidiana em xyz	52289	418313	475.211216084949417
Euclidiana em xyz *compensada*	54766	438129	475.211216084949417
Diagonal em xy	49911	399289	475.211216084949417
Diagonal em xyz	47248	377985	475.211216084949417

Estado Inicial: X = 141 ; Y = 512

Estado Final (Objectivo): X = 800 ; Y = 128

Heurística	**Nós Expandidos**	**Nós Gerados**	**Custo**
Euclidiana em xy	253444	2026791	1106.076564116243
Euclidiana em xyz	245943	1966777	1106.076564116243
Euclidiana em xyz *compensada*	253127	2024267	1106.076564116243
Diagonal em xy	209335	1674681	1106.076564116243
Diagonal em xyz	199855	1598841	1106.1665836885163

Neste último exemplo colocamos os tempos pois é o único onde realmente se nota uma diferença significativa da influência dos tempos de processamento das várias heurísticas.

Estado Inicial: X = 420 ; Y = 650
Estado Final (Objectivo): X = 999 ; Y = 10

Heurística	**Nós Expandidos**	**Nós Gerados**	**Custo**	**Tempo (ms)**
Euclidiana em xy	501794	4013243	1372.075576188206	34892 ms
Euclidiana em xyz	483311	3865802	1372.075576188206	33111 ms
Euclidiana em xyz *compensada*	15949321	127317972	1372.075576188206	569351 ms
Diagonal em xy	415117	3319827	1372.075576188206	27376 ms
Diagonal em xyz	394243	3153045	1372.075576188206	25267 ms

4 Conclusão

Nos testes em que foram comparados os algoritmos Depth First, Breadth First e Uniform Cost, pode-se ver que o número de nós expandidos e gerados é inferior quando é usada a procura em profundidade. Já o custo é superior ao custo dos outros algoritmos. Este resultado era o esperado uma vez que o DepthFirst não é óptimo, ou seja, a solução encontrada nem sempre é a de menor custo, enquanto que os outros dois são. Com algoritmo de procura AStar (A*) tanto o número de nós gerados e expandidos, como o tempo de execução, é menor quando comparado com os restantes algoritmos, *desde que se use um heurística que providencie um ganho na redução do número de nós que necessitam de ser expandidos até adquirir a solução optima que compense o overhead necessario para computar a heurística.*. Conforme a heurística, estes valores podem variar. Logo, melhores heurísticas podem produzir melhores resultados. Como é possível ver nas tabelas apresentadas, as heurísticas que não são admissiveis podem nem sempre devolver o custo menor.