Instituto Superior Técnico

IASD

Inteligência Artificial e Sistemas de Decisão

Laboratório Nr. 1

Autores:
Eduardo Lima Simões da Silva, 69916
José Pedro Braga de Carvalho Vieira Pereira, 70369

17 de Outubro de 2014

Contents

1 Introdução e Identificação do Problema

Este relatório apresenta a resolução do problema proposto como o primeiro laboratório de IASD, onde se pretende solucionar um labirinto. Para tal, foram desenvolvidos dois programas em Python (versão 3.4.1), usando diferentes métodos de procura.
Com vista a solução de um qualquer puzzle, introduzido sobre a forma de input através de um ficheiro de texto, foi inicialmente desenvolvido um método de procura não informado (Breadth-first search) e posteriormente um método de procura informado (Greedy best-first search), cuja função heuristica usada é caracterizada por $h(n)$, definido pela distância da posição actual à solução em linha recta. Por ultimo, o metodo utilizado foi caracterizada por $f(n) = h(n)+g(n)$, onde $h(n)$ já foi definido anteriormente e $g(n)$ é o custo do percurso já efectuado até ao nodo n.
Com os resultados obtidos, sob a forma de conclusão, foi possível comparar os métodos de procura usados, tanto em termos de tempo de computação como nodos gerados e tamanho do percurso.
O problema proposto consiste num labirinto que contém diversas portas, que podem estar abertas ou fechadas. No labirinto existem também interruptores que através da acção de "Push"(P) permitem alterar o estado das portas que lhe correspondam (e.g. interruptor A muda o estado de todas as portas A). O objectivo final é chegar a uma meta definida no mapa dado através de uma sequencia de acções, Left (L), Right (R), Down (D), Up (U)e Push (P) partindo de um ponto inicial também este definido no mapa dado.
O estado inicial, que será o primeiro nodo, é definido pelas coordenadas do ponto inicial (y,x), pelo caminho já percorrido ("string" com o conjunto de acções realizadas, inicialmente vazia) e pelo mapa do labirinto que é definido por um conjunto de números como explicado no enunciado (matriz: linhas x colunas).
Cada iteração consiste numa sequência de passos:

1. Expanção dos nodo/nodos escolhidos. consoante o método de pesquisa escolhido, acrescentando a respectiva acção ao caminho percorrido(path cost), actualizando as actuais coordenadas do nodo e renovando o mapa caso um botão seja premido;

2. Comparação dos nodos recentemente expandidos com os que já foram explorados (Closed List) e com os gerados anteriormente (Open List) de maneira a eliminar nodos repetidos(com algumas restrições);

3. Verificação se a meta(Goal state) foi atingida.

2 Identificação dos Algoritmos

No caso da procura não informada, foi utilizado o método de Breath-first search, com um nível de profundidade. Ou seja, a cada iteração, todos os nodos são expandidos (começando pelo que foi criado à mais tempo). Foi escolhido este método pois trata-se de um método completo que garante uma solução óptima, sendo o número de iterações igual ao path cost, e o tempo computacional/espaço alocado proporcional ao número de hipóteses a esse grau de profundidade (nodos expandidos).
Para a procura informada, foi utilizado o método de Greedy best-first search e o método A*, que a cada iteração seleciona apenas o nodo com o melhor valor na função heurística para expandir. Mantendo a eliminação dos nodos repetidos, eliminando sempre os que possuem maior caminho percorrido, garantiu-se desta maneira sempre um método completo. O método Greedy best-first apesar de ser um método não optimo pela sua heuristica, apesar de nos testes apresentados a solução optima ter sido obtida. O metodo A* implementado é optimo(garante

a obtenção da solução optima) pois ao usar a distancia em linha recta ao objectivo na sua heuristica nunca sobreestima a função de custo sendo por isso admissivel e por consequência optimo. Com estes métodos, é reduzido o espaço alocado, tendo em conta que várias hipóteses são evitadas por aumentarem a função heurística. Contudo o tempo computacional por vezes é superior, visto que por norma gera mais iterações, tal como é verificado na proxima secção do relatório.
A função heurística é a componente fundamental deste método. Assim sendo, foram implementadas duas funções heurísticas que desencadearam tempos computationais e espaços alocados ligeiramente diferentes.
Numa primeira abordagem(Greedy best-first search), a função heurística é simplesmente definida pela distância à posição final ($h(n)$), sendo o nodo a expandir aquele que apresentar menor distância. Posteriormente, com o intuito de melhorar o algoritmo, a função heurística foi definida pela soma dessa distância ao caminho já percorrido ($f(n) = g(n) + h(n)$), método A*.
O algoritmo desenvolvido em Python, para qualquer dos métodos desenvolvidos, separa claramente a parte dependente (que depende deste problema em específico) da parte independente (que apenas engloba a noção geral do algoritmo que pode ser usado para qualquer tipo de problema). A função main do algoritmo constitui a parte independente onde é possível verificar com clareza os passos de cada iteração, esta irá invocar diversas funções, desta vez dependentes, visto conterem partes especificas do algoritmo directionadas para o problema em questão.

3 Análise de Resultados e Conclusão

Por ultimo os resultados de tempo de computação, e do numero de nodos expandidos e criados são apresentados na seguinte tabela onde são comparados, e analisados de seguida:

Método	Tempo (s)			Profundidade			Nodos Exp.			Nodos gerados		
Mapa	1	6	7	1	6	7	1	6	7	1	6	7
Breadth-first (a)	0.019	0.781	0.081	50	30	140	156	1082	363	157	1106	364
Greedy best-first(b)	0.093	0.127	0.451	62	30	140	147	108	348	148	135	350
A* (c)	0.087	10.350	0.325	50	30	140	146	619	355	147	719	356
Comp. (a,c) (%)	-78.16	-92.45	-75.08	0.0	0.0	0.0	6.85	74.80	2.25	6.80	53.82	2.25

É então possivel verificar que na maioria das situações os métodos de pesquisa informado são melhores que os não informado em especial no que diz respeito ao numero de nodos expandidos e gerados onde, como seria de esperar, os metodos de pesquisa informada geram e expandem menos nodos, requerendo por isso menos memória. Outro aspecto que merece atenção é o facto de o tempo de computação ser bastante superior nos métodos de pesquisa informada. Tal facto deve-se maioritariamente ao processo iterativo onde no metodo de pesquisa informado apenas um nodo(o de menor valor na heuristica) será expandido, enquanto que no não informado todos os nodos gerados na iteração anterior são expandidos.
Podemos então concluir que o objectivo deste trabalho laboratorial foi atingido com sucesso. Neste trabalho foram implementados três métodos de pesquisa diferente, um não informado e dois informados com características diferentes. Foi também verificado que tanto o Breath-first search como o método A* são optimos e completos e o método Greedy best-first search apesar de não garantir a solução optima é completo. Existem no entanto alguns aspectos que poderão ser melhorados no futuro, nomeadamente a criação de uma função heuristica mais eficiente e inteligente, que levaria a termos menos nodos gerados, contudo para isso ser possivel, neste tipo de labirintos simples iia-se abdicar de tempo de computação.

References

[1] Luís Custódio, Rodrigo Ventura, North tower – ISR / IST, 2014, *Lecture notes - Course Artificial Intelligence and Decision Systems*

[2] Stuart Russell, Peter Norvig , 2003, Prentice Hall, Second Edition, *Artificial Intelligence: A Modern Approach*