

Universidade do Porto
Faculdade de Ciências
Departamento de Ciências dos Computadores

Relatório
Buscas Informadas e Não Informadas
Sistemas Inteligentes

Professor
Inês Dutra
ines@dcc.fc.up.pt

Aluno
Rui d'Orey
c0707046@alunos.dcc.fc.up.pt

Porto
Março de 2013

Índice

Artigo I.	Introdução	1
Artigo II.	Objetivos	1
Artigo III.	Fundamentação Teórica	2
Secção 3.01	Algoritmos de Busca Não Informada	2
Secção 3.02	Algoritmos de Busca Informada	4
Artigo IV.	Resultados	5
Artigo V.	Considerações Finais	6
Artigo VI.	Referências Bibliográficas	6

Artigo I. Introdução

Neste relatório são apresentadas e descritas as estratégias de busca em profundidade, em largura, em profundidade iterativa, gulosa e A*.

Um problema de busca caracteriza-se por ser, no contexto em que o estudamos, **um problema computacional** cujo objectivo passa por (tendo um conjunto de regras -uma função sucessória- para aplicar a um determinado estado inicial) gerar estados intermedios, com objectivo de chegar a um estado final.

Artigo II. Objetivos

Este trabalho teve como objetivo geral analisar várias estratégias de busca e como chegam a um determinado estado final partindo de um estado inicial. Será estudado se esse estado final pode ser atingido e quais formas são mais ou menos eficazes a nível de tempo e recursos computacionais usados.

Pode-se especificar que os recursos a estudar serão a memória consumida(num contexto de abstração será o numero de nós gerados, visitados e profundidade da arvore) e tempo de processamento (milisegundos que cada algoritmo necessita para, com a configuração dada, encontrar a solução).

Com estes dados, será feita uma comparação entre as várias estratégias de busca. Através da mesma tentar-se-à categorizar para que tipo de problema melhor se enquadra determinada estratégia.

Artigo III. Fundamentação Teórica

Secção 3.01 Algoritmos de Busca Não Informada

Um algoritmo de busca não informada consiste num algoritmo de busca que não efectua nenhum tipo de avaliação – heurística - acerca da melhor forma de escolher a prioridade dos nós a avaliar que seja baseada nas especificidades do problema em questão.

O método e critérios usados para seleccionar a ordem da pesquisa da solução são sempre os mesmos, independentemente do problema em questão.

Busca em Profundidade

O algoritmo de busca em profundidade procura a solução através da pesquisa, nos nós que compoem o conjunto de estados que são possíveis gerar através do nó pai, de um estado identico ao final.

A seleção do nó alcançavel mais longe do nó raiz é o critério que é usado para seleccionar o próximo nó a analisar.

O facto de se seguir um caminho à vez pode tornar a pesquisa muito demorada, se o caminho escolhido for distante da solução. E se existir um caminho pouco profundo para a solução que não foi o escolhido inicialmente.

Para tornar a função mais eficiente pode-se por exemplo, verificar se o próximo nó a verificar já foi visitado. Se isso ocorrer, ignora-se o nó.

Outro tipo de optimizações podem ser experimentadas. Efectuar uma busca em simultâneo no sentido inverso (do nó final tentar encontrar o nó inicial), comparando com os nós já encontrados nas duas pesquisas que estão a ser executadas (da origem para destino e do destino para a origem), também poderá contribuir para melhoria do tempo de reposta.

Não encontra necessariamente soluções optimas.

Busca em Largura

Este algoritmo difere do anterior na forma como a seleção do próximo nó a avaliar é feita.

Neste caso o nó não visitado com menor profundidade é selecionado.

Como vários caminhos são testados em simultâneo, o risco de se seguir um caminho muito grande é reduzido em comparação com a pesquisa em profundidade pois se a solução poder ser encontrada numa profundidade baixa, não se corre o risco de ter de analisar desnecessariamente a fundo um caminho muito distante antes de se analisar um mais pequeno.

No entanto, em casos em que exista um número muito grande de estados necessários para atingir o estado final, ou em que cada nó tenha muitas expansões, a árvore aumenta em largura proporcionalmente a esses casos, aumentando em muito o tempo de pesquisa.

A solução encontrada é sempre ótima.

Busca em Profundidade Iterativa

O algoritmo de busca em profundidade iterativa é um algoritmo de pesquisa em profundidade que assegura que a profundidade dos nós analisados é controlada. Enquanto uma solução não for encontrada é efectuada uma busca em profundidade até a uma determinada profundidade. Se essa busca não tiver resultados, é reiniciada a pesquisa em profundidade mas com um acréscimo no limite de profundidade máxima. E assim sucessivamente até ser encontrada solução.

A solução encontrada é sempre ótima.

Secção 3.02 Algoritmos de Busca Informada

Ao contrário dos algoritmos anteriores, os algoritmos de busca informada efectuem uma avaliação de qual nó será o próximo a ser testado mais adaptada à especificidade do problema a resolver – uma avaliação com base em heurísticas que são particulares do problema a avaliar.

Busca Gulosa

O algoritmo de busca gulosa aplicado neste caso com a função Manhattan, selecciona o próximo nó a avaliar de acordo com um valor atribuído a cada nó descoberto que constitui uma estimativa de quantos movimentos faltarão hipoteticamente para atingir uma solução final.

Na implementação em questão do algoritmo, o nó a ser avaliado será um qualquer que foi gerado, desde que não tenha sido visitado e tenha o menor valor de “heurística”, aproximando-se de algoritmos como o best-first.

O valor calculado só tem em conta a distância em movimentos do tabuleiro actual para o final e não a distância percorrida até ao tabuleiro actual.

Desse modo a solução obtida não será garantidamente ótima pois poderá ser dada preferência a nós mais profundos mas com tabuleiros mais aproximados do final.

Busca A-Estrela

O algoritmo A-Estrela tem regras análogas ao algoritmo guloso, exceptuando o cálculo da heurística.

Neste caso, o valor calculado não só tem em conta a distância em movimentos do tabuleiro actual para o final mas também a distância percorrida do estado inicial até ao estado actual.

O valor calculado para heurística é então o valor heurístico do algoritmo anterior mas ordenado pela menor profundidade. É garantido que a melhor escolha é feita.

Desse modo a solução obtida será ótima.

Artigo IV. Resultados

	Prof.	Larg.	Prof.It.	Guloso	A*
Tempo	276564ms (276s)	9627ms (9s)	2486ms (2s)	1344ms (1s)	19ms (0s)
Profund.	94422	20	20	114	20
Nós Visitados	102314	276995	649380	9605	76
Nós Gerados	288417	739236	1022956	28352	225

As experiencias correram numa máquina intel core i5 1,7Ghz, com 8 Gbytes de memoria.

A linguagem escolhida para a implementação das estrategias foi Java.

Artigo V. Considerações Finais

Neste trabalho, fiz a implementação e testei várias estratégias de busca informadas e não informadas.

Os resultados confirmam que as estratégias de busca não informadas poderão ser mais ou menos rápidas consoante a localização do nó solução. Independentemente disso, a estratégia busca por profundidade, além de não garantir solução ótima, pode obter soluções muito profundas e demoradas.

A estratégia informada que também não garante resultados ótimos, a busca gulosa, é bastante mais rápida que a anterior e a quantidade de memória gerada é bastante inferior às restantes estratégias não informadas.

A busca por profundidade iterativa obteve um resultado ótimo rapidamente, mas para isso necessitou de bastante memória.

A busca por profundidade será sempre ótima e quanto menos profunda for a árvore, menor será o tempo para obter uma solução.

A melhor estratégia para este caso foi a A-estrela, cujo tempo, solução e memória ocupada foram os melhores, destacando-se pela diferença de memória desperdiçada, em comparação com todas as outras estratégias.

Poderá-se concluir que, pelo menos em determinados casos, uma estratégia informada é mais eficaz que uma não informada.

Artigo VI. Referências Bibliográficas

DUTRA, I., Estratégias de busca, 2013
<http://www.dcc.fc.up.pt/~ines/aulas/1213/SI/buscas1.pdf>

DUTRA, I., Estratégias de Busca: Métodos Informados, 2013
<http://www.dcc.fc.up.pt/~ines/aulas/1213/SI/informados.pdf>