



Para você fazer

Para resolver o exercício, suponha a seguinte configuração inicial

3	6	2
7	5	4
1	9	8

Busca em Espaço de Estados. Algoritmo A*

Imagine o quebra cabeça 8 (ou 15), também conhecido como TARKIN Trata-se de um dispositivo de plástico, com espaço para 9 pedras (3x3). Existem apenas 8 pedras, um espaço reservado fica vazio. Os 4 vizinhos do espaço podem se mover sobre ele, e tudo se dá como se o espaço se movesse ao longo do tabuleiro.

Trata-se um problema de busca em espaço de estados. Conforme a definição um problema deste tipo pode ser posto como $P=I,O,C$

Neste caso I é a configuração inicial do tabuleiro. Vamos representá-la como um vetor numérico de 1 a 9, onde o espaço está representado pelo 9. Assim, uma determinada configuração como 3, 5, 4, 1, 2, 6, 7, 9, 8 deve ser entendida como

3	5	4
1	2	6
7		8

A lista dos 4 operadores refere-se aos pseudo-movimentos do espaço e são

- o espaço move-se para cima
- o espaço move-se para a direita
- para baixo
- para a esquerda

A configuração final procurada (alvo) é: 1 2 3 / 4 5 6 / 7 8 9

Apenas metade dos possíveis estados obtidos combinando-se as pedras pode ser resolvido. O número de inversões deve ser par. Uma explicação deste fato pode ser encontrada no excelente livro "O ultimo Teorema de Fermat"

A função h' será um estimador do esforço necessário para chegar à condição final. Podem-se usar diversas funções estimadoras. Quanto maior a qualidade de h' mais eficiente será a busca.

Por simplicidade, usaremos uma função quase trivial, que conta a quantidade de movimentos necessários para chegar até a configuração final.

Por exemplo, o h' do estado (2 4 6 / 8 3 1 / 5 9 7) é: 15.

A escolha do próximo nodo é: mínimo ($h' + \text{profundidade do nodo}$)

Nodos já existentes na árvore devem ser desprezados.

Em caso de empate, expandir o nodo cujo número seja menor.

Exemplo

- Seja a configuração $n_1 = 4\ 3\ 5\ /\ 7\ 1\ 9\ /\ 8\ 2\ 6$. Na profundidade 0 seu h' é: 12 ($1+1+2+1+2+1+1+2+1$). Total do nodo = 12. Este é o nodo n_1 .
- O nodo n_2 é o filho 1 de n_1 que é: 4 3 9 / 7 1 5 / 8 2 6, e cujo h' é 12 e a profundidade é 1. Total=13. Comparando n_1 (12+0) com n_2 (12+1) o nodo n_1 é menor e deve ser expandido. O próximo operador seria o 2, que é inválido. Com isso o nodo 3 será o filho de n_1 com operador 3 e fica:
- O nodo n_3 é o filho de n_1 , com $op=3$ e $n_3=4\ 3\ 5\ /\ 7\ 1\ 6\ /\ 8\ 2\ 9$. Este nodo tem $h=10$, e $p=1$, logo seu custo é $10+1=11$
Comparando os custos temos $n_1=12$, $n_2=13$ e $n_3=11$. O escolhido é o n_3 .
- Com isso, o n_4 será gerado a partir de n_3 , com operador=4 (os operadores 1, 2 e 3 são inválidos. Porque ?. Note que n_3 passa a ter custo ∞ pois esgotou toda a sua capacidade de gerar filhos.
 n_4 =filho de n_3 , $op=4$, que dá $n_4=4\ 3\ 5\ /\ 7\ 1\ 6\ /\ 8\ 9\ 2$ ($h=12$, $p=2$, $T=14$)
Comparando, $n_1=12$, $n_2=13$, $n_3=\infty$, $n_4=14$, e o nodo escolhido é n_1 .
- Com isso, o n_5 , será gerado a partir de n_1 com operador 4, e fica $n_5 = 4\ 3\ 5\ /\ 7\ 9\ 1\ /\ 8\ 2\ 6$ ($h=14$, $p=1$, $T=15$). Note que $n_1=\infty$.
Comparando $n_1=\infty$, $n_2=13$, $n_3=\infty$, $n_4=14$, $n_5=15$ e o escolhido é o n_2 .
- n_6 será gerado a partir de n_2 com $op=4$ (1, 2 e 3 inválidos). Fica $n_6=4\ 9\ 3\ /\ 7\ 1\ 5\ /\ 8\ 2\ 6$ ($h=12$, $p=2$, $T=14$). n_2 passa a ficar com ∞ .
Comparando $n_1=\infty$, $n_2=\infty$, $n_3=\infty$, $n_4=14$, $n_5=15$, $n_6=14$. O escolhido é n_4 (Note que como houve empate entre n_4 e n_6 , escolhe-se n_4).
- n_7 a partir de n_4 , $op=1$, que dá $n_7=4\ 3\ 5\ /\ 7\ 9\ 6\ /\ 8\ 1\ 2$ ($h=14$, $p=3$, $T=17$) Comparando $n_1=n_2=n_3=\infty$, $n_4=14$, $n_5=15$, $n_6=14$, $n_7=17$. O escolhido é n_4 com $op=4$, pois os operadores 2 e 3 são inválidos.
- n_8 a partir de n_4 , $op=4$ que dá $n_8=4\ 3\ 5\ /\ 7\ 1\ 6\ /\ 9\ 8\ 2$ ($h=12$, $p=3$, $T=15$) e n_4 fica ∞ .
Comparando $n_1=n_2=n_3=n_4=\infty$, $n_5=15$, $n_6=14$, $n_7=17$, $n_8=15$. O escolhido é n_6 que tem operadores 1 e 2 inválidos.
- n_9 deriva de n_6 , $op=3$, que dá $n_9=4\ 1\ 3\ /\ 7\ 9\ 5\ /\ 8\ 2\ 6$ ($h=10$, $p=3$, $T=13$)
Comparando $n_1 = n_2 = n_3 = n_4 = \infty$, $n_5 = 15$, $n_6 = 14$, $n_7 = 17$, $n_8 = 15$, $n_9 = 13$. O escolhido é n_9 que tem operador 1 inválido.
- n_{10} vem de n_9 , $op=2$, que dá $n_{10}=4\ 1\ 3\ /\ 7\ 5\ 9\ /\ 8\ 2\ 6$ ($h=8$, $p=4$, $T=12$)
- ...E assim por diante...

Pergunta Qual é o 15 nodo da árvore ?

Resposta

Implementação

Recomenda-se a implementação deste algoritmo. Se for feita um projeto modular, outros problemas poderão ser resolvidos usando o mesmo produto, com mudanças apenas cosméticas.

Uma proposta de implementação sugere usar um BD relacional contendo uma estrutura de dados formada por:

campo	tipo	tamanho
número nodo	N	8
configuração	variável	
profundidade	N	4
hlinha	N	2
endereço do pai	N	8
operador usado	N	2

Com índices por configuração (para prevenir duplos) e por $k_1 \times P_n + k_2 \times h'$ para recuperar o nodo candidato a expandir. O uso de um BD ao invés da memória tranquiliza quanto ao crescimento exponencial da árvore.

As funções a desenvolver e suas funcionalidades são:

Função	o que deve fazer
inicialização	estabelecer os parâmetros da rodada, criar o BD e os índices capturar a configuração inicial e final e a heurística
viabilidade	análise se esta proposta inicial tem solução
principal	loop: acha o menor, avalia se fim, expande o nodo, inclui no BD
mostracaminho	ao achar a solução recupera e imprime o caminho
hlinha	calcula e devolve a heurística
achafilho	a partir do pai e do operador calcula o filho

