Agentes Cognitivos Adaptativos

Lista de exercícios sobre Search

Obs.: esta lista não vale nota. Serve apenas para treinar a matéria da prova!

# **A\* - Problema do metrô de Paris**

Suponha que queremos construir um sistema para auxiliar um usuário do metrô de Paris a saber o **trajeto mais rápido** entre a estação onde ele se encontra e a estação de destino. O usuário tem um painel com o mapa, podendo selecionar a sua estação de destino. O sistema então acende as luzes sobre o mapa mostrando o melhor trajeto a seguir (em termos de quais estações ele vai atravessar., e quais as conexões mais rápidas a fazer – se for o caso). Para facilitar a vida, consideramos apenas 4 linhas do metrô.

Considere que:

* a distância em linha reta entre duas estações quaisquer é dada pela tabela 1 e a distância real é dada pela tabela 2.
* a velocidade média de um trem é de 30km/h;
* o tempo gasto para trocar de linha dentro de mesma estação (fazer baldeação) é de 4 minutos.

# **Questões:**

# **Formule** este problema em termos de estado inicial, estado final, operadores e função de avaliação para **Busca heurística com A\*.**

1. **Descreva o funcionamento** do algoritmo em linguagem corrente (5 linhas).
2. **Desenhe a árvore de busca** desenvolvida pelo A\* para o seguinte caso:

Ei = estação 6 linha azula

Ef = estação 13 linha vermelha

**Dados do problema:**

**Tabela1:** distâncias **diretas** entre as estações do Metrô de Paris.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **E1** | **E2** | **E3** | **E4** | **E5** | **E6** | **E7** | **E8** | **E9** | **E10** | **E11** | **E12** | **E13** | **E14** |
| **E1** | - |  10 | 18,5 | 24,8 | 36,4 | 38,8 | 35,8 | 25,4 | 17,6 | 9,1 | 16,7 | 27,3 | 27,6 | 29,8 |
| **E2** |  | - | 8,5 | 14,8 | 26,6 | 29,1 | 26,1 | 17,3 | 10 | 3,5 | 15,5 | 20,9 | 19,1 | 21,8 |
| **E3** |  |  | - | 6,3 | 18,2 | 20,6 | 17,6 | 13,6 | 9,4 | 10,3 | 19,5 | 19,1 | 12,1 | 16,6 |
| **E4** |  |  |  | - | 12 | 14,4 | 11,5 | 12,4 | 12,6 | 16,7 | 23,6 | 18,6 | 10,6 | 15,4 |
| **E5** |  |  |  |  | - | 3 | 2,4 | 19,4 | 23,3 | 28,2 | 34,2 | 24,8 | 14,5 | 17,9 |
| **E6** |  |  |  |  |  | - | 3,3 | 22,3 | 25,7 | 30,3 | 36,7 | 27,6 | 15,2 | 18,2 |
| **E7** |  |  |  |  |  |  | - | 20 | 23 | 27,3 | 34,2 | 25,7 | 12,4 | 15,6 |
| **E8** |  |  |  |  |  |  |  | - | 8,2 | 20,3 | 16,1 | 6,4 | 22,7 | 27,6 |
| **E9** |  |  |  |  |  |  |  |  | - | 13,5 | 11,2 | 10,9 | 21,2 | 26,6 |
| **E10** |  |  |  |  |  |  |  |  |  | - | 17,6 | 24,2 | 18,7 | 21,2 |
| **E11** |  |  |  |  |  |  |  |  |  |  | - | 14,2 | 31,5 | 35,5 |
| **E12** |  |  |  |  |  |  |  |  |  |  |  | - | 28,8 | 33,6 |
| **E13** |  |  |  |  |  |  |  |  |  |  |  |  | - | 5,1 |
| **E14** |  |  |  |  |  |  |  |  |  |  |  |  |  | - |

**Tabela2:** distâncias **reais** entre as estações do Metrô de Paris.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **E1** | **E2** | **E3** | **E4** | **E5** | **E6** | **E7** | **E8** | **E9** | **E10** | **E11** | **E12** | **E13** | **E14** |
| **E1** | - |  10 |  |  |  |  |  |  |  |  |  |  |  |  |
| **E2** |  | - | 8,5 |  |  |  |  |  | 10 | 3,5 |  |  |  |  |
| **E3** |  |  | - | 6,3 |  |  |  |  | 9,4 |  |  |  | 18,7 |  |
| **E4** |  |  |  | - | 13 |  |  | 15,3 |  |  |  |  | 12,8 |  |
| **E5** |  |  |  |  | - | 3 | 2,4 | 30 |  |  |  |  |  |  |
| **E6** |  |  |  |  |  | - |  |  |  |  |  |  |  |  |
| **E7** |  |  |  |  |  |  | - |  |  |  |  |  |  |  |
| **E8** |  |  |  |  |  |  |  | - | 9,6 |  |  | 6,4 |  |  |
| **E9** |  |  |  |  |  |  |  |  | - |  | 12,2 |  |  |  |
| **E10** |  |  |  |  |  |  |  |  |  | - |  |  |  |  |
| **E11** |  |  |  |  |  |  |  |  |  |  | - |  |  |  |
| **E12** |  |  |  |  |  |  |  |  |  |  |  | - |  |  |
| **E13** |  |  |  |  |  |  |  |  |  |  |  |  | - | 5,1 |
| **E14** |  |  |  |  |  |  |  |  |  |  |  |  |  | - |

**Mapa do metrô de Paris.**



1. **Otimização -** **Subida da Encosta**

Uma empresa deseja montar uma rede de computadores interna, instalando máquinas em várias salas (12 máquinas ao todo).  Cada computador estará ligado a dois outros, com exceção do primeiro e do último  (que estão ligados apenas a um outro computador). Nem todas as combinações de ligações entre computadores são possíveis. A tabela 3 abaixo indica as possibilidades de ligação (o traço (-) indica que não há conexão possível entre os computadores indicados). A empresa deseja fazer essas ligações de forma a economizar o cabo (em metros). Estamos diante de um problema de **otimização.**

O operador considerado para gerar os sucessores do estado corrente é apenas a permutação da ordem atual das ligações entre computadores dois a dois, sem testar todas as combinações em uma mesma iteração. Por exemplo, dado o estado inicial:

**Ei (C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12)**

As permutações em uma iteração seriam:

E1 (**C2, C1**, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12)

E2 (C1, **C3, C2**, C4, C5, C6, C7, C8, C9, C10, C11, C12)

E3 (C1, C2, **C4, C3**, C5, C6, C7, C8, C9, C10, C11, C12) etc...

A distância direta entre os pontos da rede são dadas na tabela abaixo.

**Tabela 3:** Distâncias entre os computadores, indicando as ligações possíveis.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **C1** | **C2** | **C3** | **C4** | **C5** | **C6** | **C7** | **C8** | **C9** | **C10** | **C11** | **C12** |
| **C1** | 0 | 30 | 84 | 56 | - | 70 | - | 75 | - | 40 | - | 10 |
| **C2** |  | 0 | 65 | - | - | - | 70 | - | - | 40 | - | - |
| **C3** |  |  | 0 | 60 | 52 | 55 | - | 135 | 143 | 48 | 25 | - |
| **C4** |  |  |  | 0 | 135 | - | - | 20 | - | - | 58 | - |
| **C5** |  |  |  |  | 0 | 70 | - | 122 | 98 | 80 | - | - |
| **C6** |  |  |  |  |  | 0 | 68 | - | 82 | 35 | - | 130 |
| **C7** |  |  |  |  |  |  | 0 | 40 | 120 | 57 | - | - |
| **C8** |  |  |  |  |  |  |  | 0 | 89 | - | 45 | - |
| **C9** |  |  |  |  |  |  |  |  | 0 | 23 | - | 68 |
| **C10** |  |  |  |  |  |  |  |  |  | 0 | 10 | - |
| **C11** |  |  |  |  |  |  |  |  |  |  | 0 | 14 |
| **C12** |  |  |  |  |  |  |  |  |  |  |  | 0 |

**Questões:**

1. Formule o problema em termos de:
	1. estado inicial;
	2. estado final;
	3. operadores (e seus custos associados);
	4. função de avaliação (h).
2. Descreva, sucintamente, o funcionamento do algoritmo (explique como você o implementaria).
3. Exemplifique o funcionamento do algoritmo em 2 iterações para

Ei= (C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12)

levando em conta os valores da função de avaliação para justificar as escolhas.

**3) MinMax**

**4) Problema com Satisfação de restrição –Alocação de recursos[[1]](#footnote-1)**Uma empresa de desenvolvimento de software deseja contratar novos progrmadores e investir mais dinheiro em hardware e software a fim de aumentar a produtividade do setor de programação. O gerente definiu que a melhoria da produtividade seria medida pelo aumento de linhas de código produzidas pelo setor. A companhia necessita de uma tabela de alocação de recursos que maximize a produtividade total.O setor de programação é subdividido em 3 divisões, e cada divisão tem restrições de como utilizar os novos recursos. Por exemplo, a divisão D1 é capaz de abrigar 0, 3, 5 ou 6 novos programadores (a organização da divisão não aceita usar 1, 2, 4, 7 ou mais novos programadores). Isto dá 4 opções de utilização de novos programadores nesta divisão. Existem apenas 4 opções diferentes para alocação de dinheiro para essa divisão (indicadas abaixo), o que resulta em 16 possíveis combinações de alocação para este exemplo. Para cada possibilidade, a empresa estimou o aumento de linhas de código produzidas pela divisão.

Determine a alocação precisa de recursos entre as divisões do setor de forma a maximizar o ganho em linhas de código do setor como um todo. A alocação deve indicar o número de programadores novos em cada divisão, bem como o total de novos investimentos por divisão.

Restrições: (1) o número total de novos programadores não pode ser maior do que 10

 (2) o valor total de novos investimentos não pode exceder $ 90.000,00

Valores permitidos para alocação em cada divisão:

Divisão 1: (1) novos programadores = {0, 3, 5, 6}

 (2) novos investimentos = {0, 20.000, 50.000, 70.000}

 (3) aumento de linhas de código = Tabela: prog *x* inv

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  InvProg | 0 | 20.000 | 50.000 | 70.000 |
| 0 | 0 | 10.000 | 20.000 | 50.000 |
| 3 | 60.000 | 20.000 | 10.000 | 40.000 |
| 5 | 20.000 | 10.000 | 30.000 | 40.000 |
| 6 | 30.000 | 10.000 | 40.000 | 30.000 |

Divisão 2: (1) novos programadores = {0, 1, 3, 4, 8}

 (2) novos investimentos = {0, 40.000, 80.000}

 (3) aumento de linhas de código = Tabela: prog *x* inv

|  |  |  |  |
| --- | --- | --- | --- |
|  InvProg | 0 | 40.000 | 80.000 |
| 0 | 0 | 50.000 | 30.000 |
| 1 | 50.000 | 40.000 | 60.000 |
| 3 | 20.000 | 30.000 | 50.000 |
| 4 | 80.000 | 90.000 | 50.000 |
| 8 | 30.000 | 40.000 | 70.000 |

Divisão 3: (1) novos programadores = {0, 4, 6}

 (2) novos investimentos = {0, 5.000, 30.000, 40.000, 50.000}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  InvProg | 0 | 5.000 | 30.000 | 40.000 | 50.000 |
| 0 | 0 | 30.000 | 50.000 | 60.000 | 30.000 |
| 4 | 10.000 | 20.000 | 30.000 | 40.000 | 50.000 |
| 6 | 20.000 | 30.000 | 40.000 | 50.000 | 60.000 |

 (3) aumento de linhas de código = Tabela: prog *x* inv

**Questões:**

**1. Formule** este problema de CSP em termos de um conjunto de variáveis e seus domínios para:

1. Busca cega: estados iniciais, estado final (teste de término), operadores.

 Como propagar restrições neste problema?

1. Busca heurística: existe alguma heurística para este problema? Qual(is)? Comente sobre cada uma delas (se existirem).
2. CSP iterativo: estado inicial, operadores, teste de término e heurística.

**2.** Descreva o funcionamento do algoritmo para o caso (c) CSP iterativo.

**Observação importante:** As tabelas de *Investimento x Programadores* NÃO devem ser usadas como dados permanentes do problema. Elas representam uma possível instância do problema, e podem mudar. Assim, elas devem ser consideradas como dados de entrada.

1. ACM Scholastic Programming Contest Finals 1993 [↑](#footnote-ref-1)