

# IC 2024.1

## Atividade 2

Aluno: Davi de Paula Coelho Raia dos Santos

### UECE - Ciência da Computação

#### Questão 1

**Estado:** Grupo de variáveis que representam as condições ou características no momento específico de um problema particular.

**Espaço de Estados:** Considerando as regras e ações possíveis, é o conjunto de todos os estados que podem ser alcançados a partir do estado inicial.

**Árvore de Busca:** Gráfico que indica a exploração dos estados possíveis de um problema particular.

**Nó:** Cada ponto ou estado na árvore de busca. Representa uma possível configuração do problema em algum ponto da solução.

**Objetivo:** Ponto final ou alvo que o agente tenta alcançar, o estado que define a solução desejada.

**Função Sucessora:** Essa função define todas as ações possíveis que podem ser realizadas num determinado estado, gera estados resultantes.

**Fator de Ramificação:** Número médio de sucessores gerados a partir de um nó (estado). Indica a quantidade de ações disponíveis em cada ponto da árvore de busca.

#### Questão 2

Item A:

- Estado Inicial: Um mapa com todas as regiões não coloridas.
- Ações: Colorir uma região com uma das quatro cores (vermelho, verde, azul, amarelo), garantindo que as regiões adjacentes tenham cores diferentes.

- Teste Meta/Objetivo: Todas as regiões do mapa estão coloridas, e regiões adjacentes têm cores diferentes.
- Função Custo ( $g(n)$ ): Número de regiões já coloridas ou a quantidade de cores usadas.
- Função de Avaliação ( $f(n) = g(n) + h(n)$ ):

$g(n)$ : Número de regiões coloridas corretamente.

$h(n)$ : Número estimado de regiões adjacentes ainda por colorir que precisam de uma cor diferente.

#### Item B:

- Estado Inicial: O macaco está no chão com duas caixas móveis separadas e as bananas suspensas no teto.
- Ações: Empurrar as caixas, empilhar as caixas, escalar as caixas, pegar bananas.
- Teste Meta/Objetivo: O macaco alcançou e pegou as bananas.
- Função Custo ( $g(n)$ ): Número de movimentos ou ações realizados pelo macaco para atingir o objetivo.
- Função de Avaliação ( $f(n) = g(n) + h(n)$ ):

$g(n)$ : Número de ações executadas (empurrar, empilhar, escalar).

$h(n)$ : Distância restante entre o macaco e as bananas (em termos de altura ou número de caixas que ele ainda precisa empilhar).

#### Item C:

- Estado Inicial: Arquivo com registros de entrada.
- Ações: Verificar um registro, processar o registro, remover ou isolar registros inválidos.
- Teste Meta/Objetivo: Encontrar e identificar o(s) registro(s) inválido(s).
- Função Custo ( $g(n)$ ): Número de registros processados até identificar o registro inválido.
- Função de Avaliação ( $f(n) = g(n) + h(n)$ ):

$g(n)$ : Número de registros já analisados.

$h(n)$ : Número estimado de registros restantes a serem verificados.

#### Item D:

- Estado Inicial: Três missionários, três canibais e o barco estão em um lado do rio.
- Ações: Mover o barco com um ou dois passageiros (missionários e/ou canibais) para o outro lado do rio.
- Teste Meta/Objetivo: Todos os missionários e canibais atravessaram o rio em segurança, sem nunca deixar missionários em menor número em qualquer lado do rio.
- Função Custo ( $g(n)$ ): Número de travessias realizadas.
- Função de Avaliação ( $f(n) = g(n) + h(n)$ ):

$g(n)$ : Número de travessias realizadas até o momento.

$h(n)$ : Número de pessoas restantes que ainda precisam atravessar o rio.

### Funções de Avaliação ( $f(n)$ )

Para todos os problemas, a função de avaliação geral é da forma  $f(n)=g(n)+h(n)$ , onde:

$g(n)$  representa o custo do caminho até o nó atual (ou seja, o número de ações realizadas até o momento).

$h(n)$  é uma estimativa do custo restante para alcançar o objetivo, ou seja, uma heurística que tenta prever a distância entre o estado atual e o estado objetivo.

A combinação de  $g(n)$  e  $h(n)$  forma a função de avaliação  $f(n)$ , usada em algoritmos de busca heurística como o A\* para determinar a melhor rota para atingir o objetivo.

## Questão 3

Busca em Largura:

Iteração 0:

- Fronteira (S)
- Custo atual 0
- Caminho Atual S

Iteração 1:

- Fronteira (A, E)
- Custo atual 0
- Caminho Atual S

Iteração 2:

- Fronteira (E, B)
- Custo atual (2, 4)
- Caminho Atual (S  $\rightarrow$  E, S  $\rightarrow$  A  $\rightarrow$  B)

Iteração 3:

- Fronteira (B, F)
- Custo atual (4, 7)
- Caminho Atual (S  $\rightarrow$  A  $\rightarrow$  B, S  $\rightarrow$  E  $\rightarrow$  F)

Iteração 4:

- Fronteira (F, C)
- Custo atual (7, 6)
- Caminho Atual (S  $\rightarrow$  E  $\rightarrow$  F, S  $\rightarrow$  A  $\rightarrow$  B  $\rightarrow$  C)

Iteração 5:

- Fronteira (C, G)

- Custo atual (6, 9)
- Caminho Atual ( $S \rightarrow A \rightarrow B \rightarrow C$ ,  $S \rightarrow E \rightarrow F \rightarrow G$ )

Iteração 6:

- Fronteira (G, D)
- Custo atual (9, 9)
- Caminho Atual ( $S \rightarrow E \rightarrow F \rightarrow G$ ,  $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D$ )

Iteração 7:

- Fronteira (D, T)
- Custo atual (9, 11)
- Caminho Atual ( $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D$ ,  $S \rightarrow E \rightarrow F \rightarrow G \rightarrow T$ )

Iteração 8:

- Fronteira (T, T)
- Custo atual (11, 12)
- Caminho Atual ( $S \rightarrow E \rightarrow F \rightarrow G \rightarrow T$ ,  $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow T$ )

Iteração 9:

- Chegou ao final
- Custo 11
- Expandido 9 nós (S, A, B, C, D, E, F, G, T)

Busca Custo Uniforme

Iteração 0:

- Fronteira (A, E)
- Custo atual 0
- Caminho Atual S

Iteração 1:

- Fronteira (B, E)
- Custo atual (2, 0)
- Caminho Atual ( $S \rightarrow A$ , S)

Iteração 2:

- Fronteira (B, F)
- Custo atual (2, 2)
- Caminho Atual ( $S \rightarrow A$ ,  $S \rightarrow E$ )

Iteração 3:

- Fronteira (C, F)
- Custo atual (4, 2)
- Caminho Atual ( $S \rightarrow A \rightarrow B$ ,  $S \rightarrow E$ )

Iteração 4:

- Fronteira (D, F)
- Custo atual (6, 2)
- Caminho Atual ( $S \rightarrow A \rightarrow B \rightarrow C$ ,  $S \rightarrow E$ )

Iteração 5:

- Fronteira (D, G)
- Custo atual (6, 7)
- Caminho Atual ( $S \rightarrow A \rightarrow B \rightarrow C$ ,  $S \rightarrow E \rightarrow F$ )

Iteração 6:

- Fronteira (T, G)
- Custo atual (9, 7)
- Caminho Atual ( $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D$ ,  $S \rightarrow E \rightarrow F$ )

Iteração 7:

- Fronteira (T, T)
- Custo atual (9, 9)
- Caminho Atual ( $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D$ ,  $S \rightarrow E \rightarrow F \rightarrow G$ )

Iteração 8:

- Fronteira (T, T)
- Custo atual (9, 11)
- Caminho Atual ( $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D$ ,  $S \rightarrow E \rightarrow F \rightarrow G \rightarrow T$ )

Iteração 9:

- Chegou ao Final (T)
- Custo de 11
- Expandido 9 Nós (S, A, B, C, D, E, F, G, T)

Busca em Profundidade

Iteração 0:

- Fronteira: (S)
- Custo atual 0
- Caminho atual: S

Iteração 1:

- Fronteira (A, E)
- Custo Atual (2, 2)
- Caminhos: ( $S \rightarrow A$  ou  $S \rightarrow E$ )

Iteração 2:

- Fronteira (B, E)
- Custo atual (4, 2)
- Caminhos ( $S \rightarrow A \rightarrow B$  ou  $S \rightarrow E$ )

Iteração 3:

- Fronteira (C, E)
- Custo atual (6, 2)
- Caminhos ( $S \rightarrow A \rightarrow B \rightarrow C$  ou  $S \rightarrow E$ )

Iteração 4:

- Fronteira (D, E)

- Custo atual (9, 2)
- Caminhos ( $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D$  ou  $S \rightarrow E$ )

Iteração 5:

- Fronteira (T, E)
- Custo atual (12, 2)
- Caminhos ( $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow T$  ou  $S \rightarrow E$ )

Iteração 6:

- Chegou ao Final (T)
- Custo de 12
- Expandidos 6 nós (S, A, B, C, D, T)

A busca em largura é um caso especial de custo uniforme em que a fila de prioridades é igual a 1, nenhum nó tem mais prioridade que o outro.

## Questão 4

Busca gulosa pela melhor escolha

Iteração 0:

- Fronteira: (S)
- Custo atual 0
- Caminho atual: S

Iteração 1:

- Fronteira (A, E)
- Custo Atual (2, 2)
- Caminhos: ( $S \rightarrow A$  ou  $S \rightarrow E$ )

Iteração 2:

- Fronteira (B, E)
- Custo atual (4, 2)
- Caminhos ( $S \rightarrow A \rightarrow B$  ou  $S \rightarrow E$ )

Iteração 3:

- Fronteira (C, E)
- Custo atual (6, 2)
- Caminhos ( $S \rightarrow A \rightarrow B \rightarrow C$  ou  $S \rightarrow E$ )

Iteração 4:

- Fronteira (D, E)
- Custo atual (9, 2)
- Caminhos ( $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D$  ou  $S \rightarrow E$ )

Iteração 5:

- Fronteira (T, E)

- Custo atual (12, 2)
- Caminhos ( $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow T$  ou  $S \rightarrow E$ )

Iteração 6:

- Chegou ao Final (T)
- Custo de 12
- Expandidos 6 nós (S, A, B, C, D, T)

Busca A\*

Iteração 0:

- Fronteira (A, E)
- Custo atual 0
- Caminho Atual S

Iteração 1:

- Fronteira (B, E)
- Custo atual(2, 0)
- Caminho atual ( $S \rightarrow A$ ,  $S \rightarrow$ )

Iteração 2:

- Fronteira (C, E)
- Custo atual (4, 0)
- Caminho Atual ( $S \rightarrow A \rightarrow B$ ,  $S \rightarrow$ )

Iteração 3:

- Fronteira (C, F)
- Custo atual (4, 2)
- Caminho Atual ( $S \rightarrow A \rightarrow B$ ,  $S \rightarrow E$ )

Iteração 4:

- Fronteira (D, F)
- Custo atual (6, 2)
- Caminho Atual ( $S \rightarrow A \rightarrow B \rightarrow C$ ,  $S \rightarrow E$ )

Iteração 5:

- Fronteira (D, G)
- Custo atual (6, 7)
- Caminho Atual ( $S \rightarrow A \rightarrow B \rightarrow C$ ,  $S \rightarrow E \rightarrow F$ )

Iteração 6:

- Fronteira (D, T)
- Custo atual (6, 9)
- Caminho Atual ( $S \rightarrow A \rightarrow B \rightarrow C$ ,  $S \rightarrow E \rightarrow F \rightarrow G$ )

Iteração 7:

- Fronteira (D, T)
- Custo atual (6, 11)
- Caminho Atual ( $S \rightarrow A \rightarrow B \rightarrow C$ ,  $S \rightarrow E \rightarrow F \rightarrow G \rightarrow T$ )

Iteração 8:

- Chegou ao Final (T)
- Custo de 11
- Expandidos 8 nós (S, A, B, C, E, F, G, T)

A busca de custo uniforme é um tipo de Busca A\* que considera a heurística nula. Contudo, apenas a busca de custo uniforme pode ser considerada um caso da busca por melhor escolha.

## Questão 5

Item A:

Sim, a busca A\* é completa e ótima, desde que as seguintes condições sejam atendidas, a heurística  $h(n)$  usada deve ser admissível, ou seja, nunca superestimar o custo real do nó atual até o objetivo e o espaço de busca deve ser finito, ou se for infinito, as distâncias acumuladas devem ser limitadas por uma função finita. A completude significa que o algoritmo A\* sempre encontrará uma solução se existir uma, e a otimalidade garante que ele encontrará a solução de custo mínimo.

Item B:

A complexidade de tempo e espaço da busca A\* pode ser alta, dependendo da natureza do problema e da qualidade da heurística, em relação ao tempo a busca A\* pode ter uma complexidade exponencial no pior caso, especialmente em problemas com muitas opções e profundidade grande. Em relação ao espaço a busca A\* precisa armazenar todos os nós gerados na fronteira, ou seja, a lista de nós ainda não explorados (fronteira) pode crescer muito. Isso pode resultar em uma complexidade de espaço também exponencial no número de nós.

A eficiência depende diretamente da qualidade da heurística  $h(n)$  utilizada. Heurísticas melhores resultam em menos nós explorados, o que reduz a complexidade.

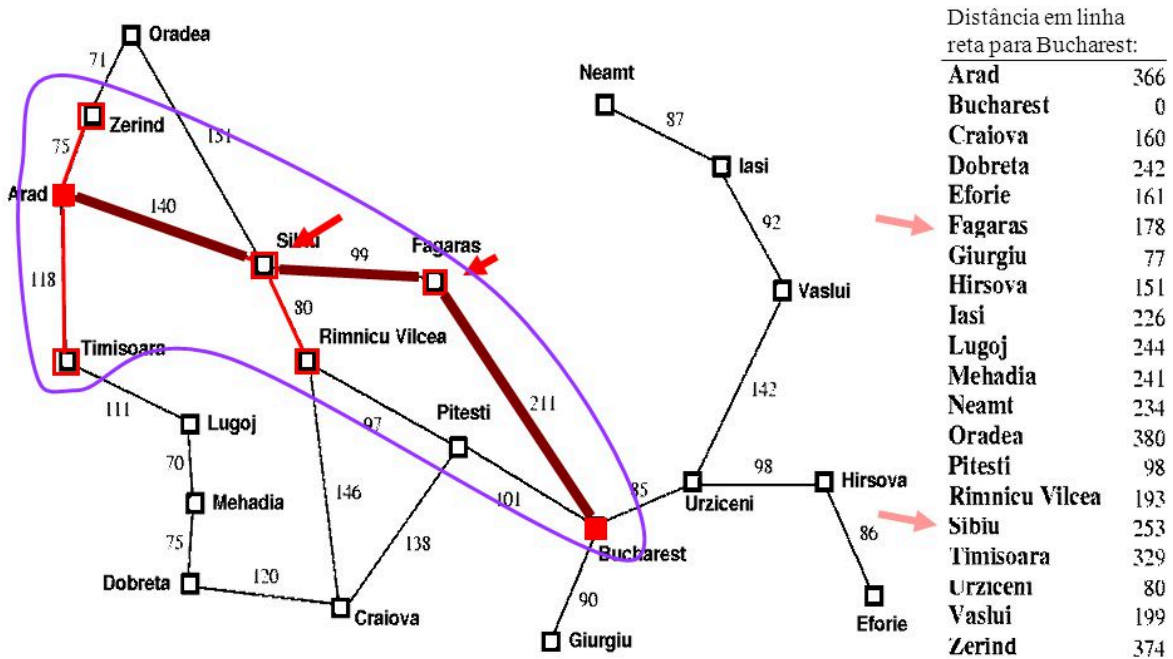
## Questão 6

Como o problema do aspirador possui dois estados possíveis por sala (limpo ou sujo) e as ações do aspirador podem ser repetidas (movimento entre salas), o uso de um grafo é mais apropriado, pois evita a exploração repetida de estados já visitados. Um algoritmo como A\* seria adequado se houver uma heurística clara para estimar a quantidade de sujeira restante. Caso contrário, algoritmos de busca não informada como busca em largura ou custo uniforme poderiam ser usados. Com base no que foi visto e implementado no primeiro trabalho um agente baseado em modelos que utilize qualquer um dos modelos propostos na resposta será mais eficiente e previsível que o agente Aleatório (Reativo ou Não) e conseguirá desempenhar melhor.



## Questão 7

### Busca Heurística - Busca pela melhor escolha - Busca Gulosa



Item A:

A busca em profundidade encontra qualquer caminho para o objetivo, mas pode não ser o mais eficiente ou direto, e pode acabar encontrando caminhos mais longos ou sub-ótimos.

Item B:

A busca em largura expande primeiro os nós mais próximos do ponto de partida. Como resultado, ela garante encontrar a solução com o menor número de arestas (caminho mais curto em termos de quantidade de nós).

Item C:

A busca de custo uniforme sempre encontra o caminho de menor custo total, independentemente do número de nós intermediários. Se todas as arestas tiverem o mesmo peso, será equivalente à busca em largura.

Item D:

A busca gulosa expande o nó mais próximo do objetivo, de acordo com a heurística, o que significa que pode seguir um caminho mais direto ao objetivo. No entanto, ela pode não garantir o caminho de menor custo, apenas o mais promissor a cada passo.

Item E:

A busca A\* combina o custo real do caminho com uma heurística de estimativa, garantindo que o caminho encontrado seja o de menor custo total, enquanto também considera a proximidade ao objetivo.

Resultados encontrados:

Profundidade:

- Custo: 733
- Nós: 8
- Caminho: Arad --> Timisoara --> Lugoj --> Mehadia --> Dobreta --> Craiova --> Pitesti --> Bucharest

Largura:

- Custo: 450
- Nós: 12
- Caminho: Arad --> Sibiu --> Fagaras --> Bucharest

Custo Uniforme:

- Custo: 418
- Nós: 15
- Caminho: Arad --> Sibiu --> Rimnicu Vilcea --> Pitesti --> Bucharest

Melhor Escolha:

- Custo: 450
- Nós: 4
- Caminho: Arad --> Sibiu --> Fagaras --> Bucharest

A\*:

- Custo: 418
- Nós: 6
- Caminho: Arad --> Sibiu --> Rimnicu Vilcea --> Pitesti --> Bucharest

As Buscas de Custo Uniforme e A\* Encontraram o melhor caminho.

Sobre o gasto de tempo, a Busca de Melhor escolha foi a mais rápida enquanto a mais lenta foi a busca de custo uniforme, isso também serve para gasto de memória, uma vez que a Busca de custo uniforme precisa armazenar mais valores.

## Questão 9:

Item A:

Esse é um caso especial de subida de encosta ou hill climbing. Quando  $k=1$ , o feixe local mantém apenas um estado de solução a cada iteração, o que corresponde à técnica de subida de encosta padrão.

Item B:

Esse caso representa a busca em feixe aleatório. Se não há limite de estados mantidos, todos os estados candidatos gerados podem ser considerados, resultando em um comportamento aleatório semelhante à busca em feixe não restringida.

Item C:

Quando a temperatura  $T$  é mantida igual a zero, a têmpera simulada se comporta como a subida de encosta, pois  $T=0$  significa que o algoritmo só aceita movimentos que melhoram a solução e rejeita todos os piores movimentos.

Item D:

Se  $T=\infty$ , o algoritmo aceita todos os movimentos, tornando-o equivalente a uma busca aleatória. A ideia da têmpera simulada é controlar a temperatura para explorar soluções ruins no início, mas quando  $T=\infty$  o controle é perdido e o processo se torna completamente aleatório.

Item E:

Se a população do algoritmo genético for reduzida a  $N=1$ , o algoritmo se transforma em uma subida de encosta ou hill climbing, porque com uma única solução por vez, não há diversidade de soluções para cruzamento, limitando-se a melhorar a única solução disponível através de mutações.