

# Aulas 2 e 3

## Algoritmos de Busca Aplicados ao Problema do Caminho Mínimo

**Prof. Dr. Peterson A. Belan**

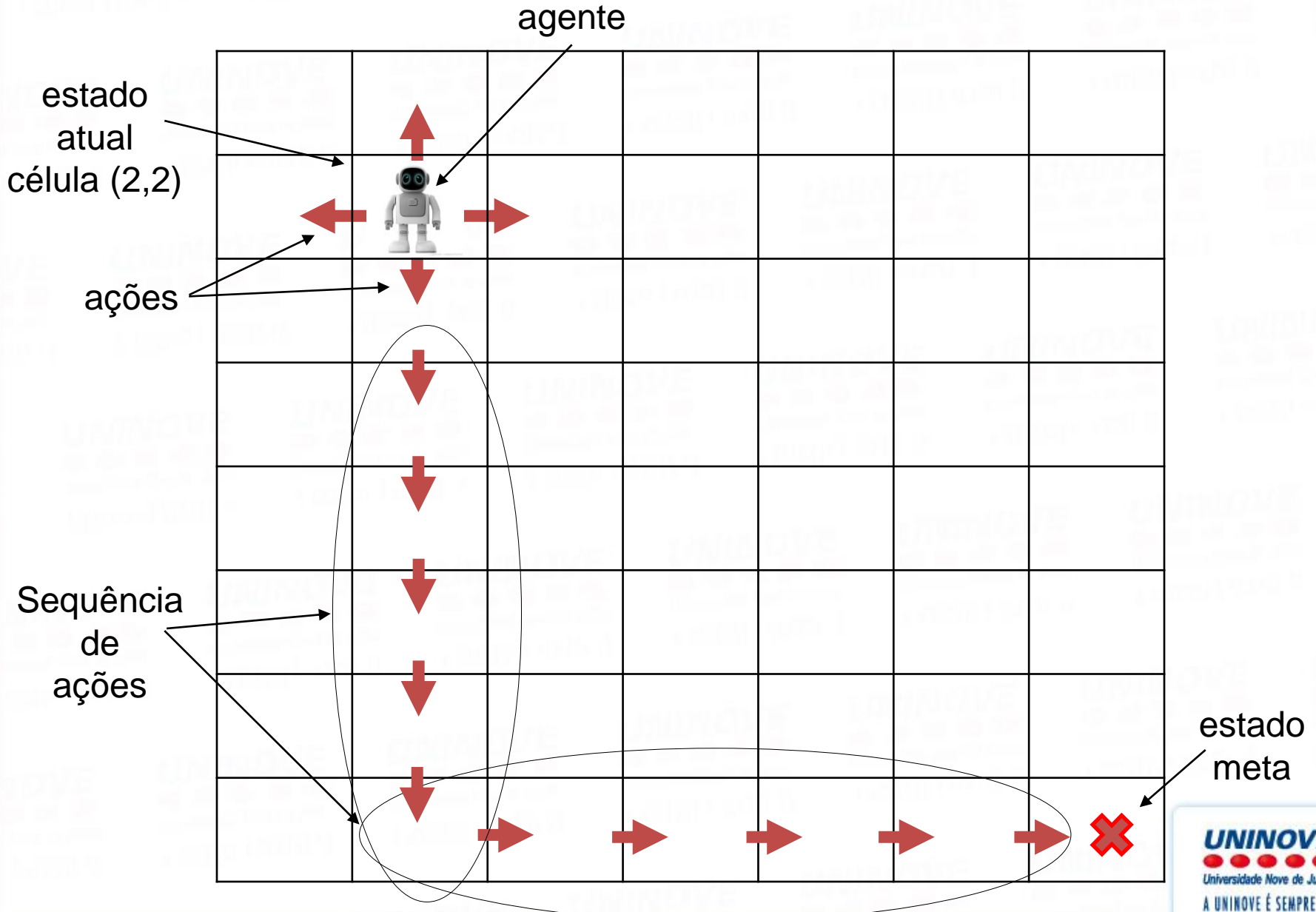
# Busca no espaço de estados

É uma das técnicas mais utilizadas para resolução de problemas em Inteligência Artificial.

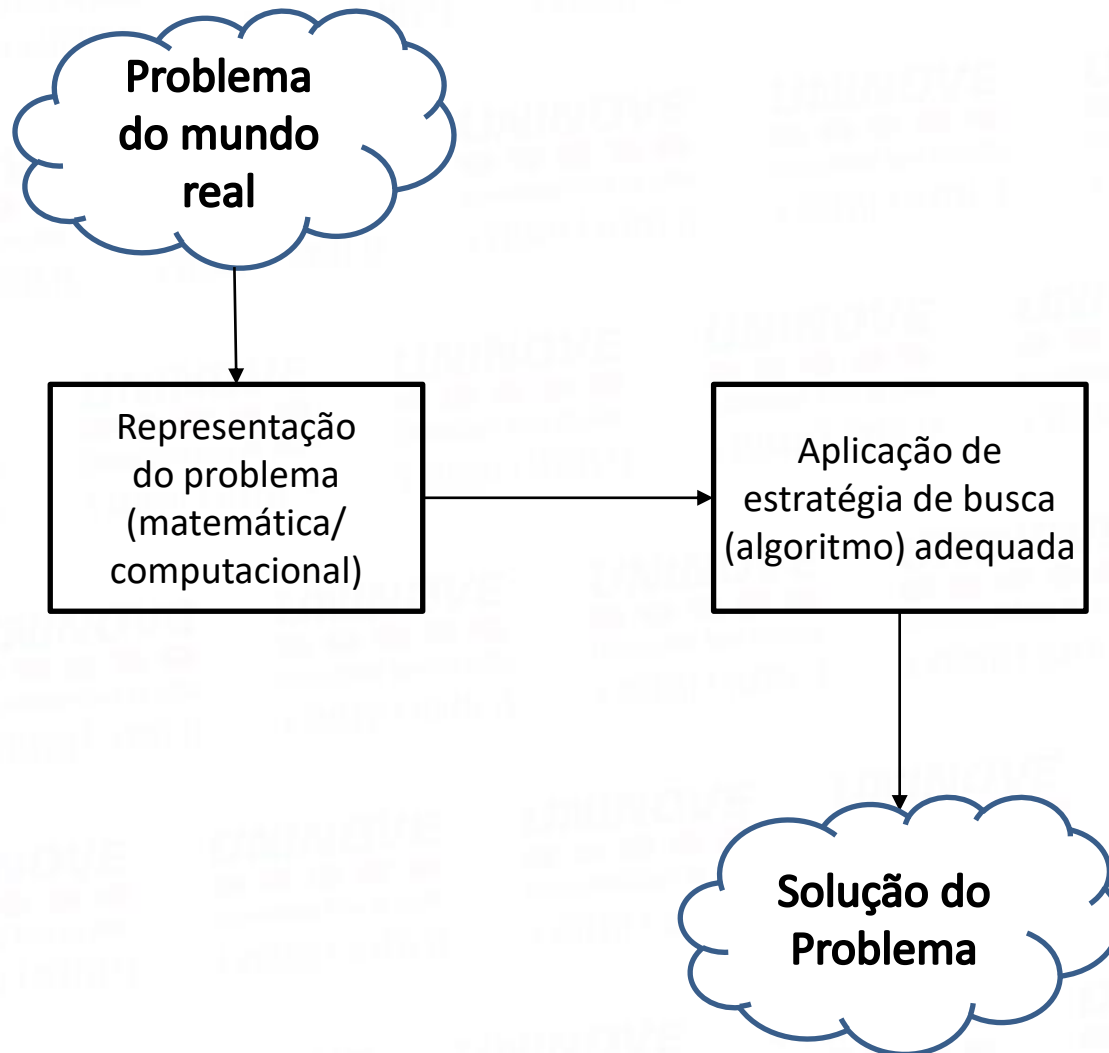
A ideia consiste em supor a existência de um **agente** capaz de **executar ações** que modificam o **estado corrente (atual)** de seu mundo.

Assim, dados um **estado inicial** representando a configuração atual do mundo do agente, um **conjunto de ações** que o agente é capaz de executar e uma descrição do **estado final (meta)** que se deseja atingir, a **solução do problema** consiste em uma **seqüência de ações** que, quando executada, leva o agente do **estado inicial** até **estado final**.

# Busca no espaço de estados



# Resolução de Problemas por meio de busca no espaço de estados



# Problema de Busca - representação

Para representar um problema como uma busca no espaço de estados, em geral, é preciso definir:

- Estados:  $s$  (o que representa?)

- Espaço de Estados:  $S$

- Estado Inicial:  $s_0$

- Estado(s) Final(is):  $G$

- Conjunto de Ações:  $A$

Uma ação  $a \in A$  pode ser descrita na forma: **ação**( $\alpha, \alpha'$ )  $\leftarrow \beta$

onde:  $\alpha$  é o estado atual,  $\alpha'$  é um estado sucessor e  $\beta$  é o conjunto de condições que devem ser obedecidas

# Exemplo (robô –slide 3)

- Estados:  $s \Rightarrow$  célula da matriz / par de inteiros (lin,col),  $1 \leq \text{lin} \leq 8, 1 \leq \text{col} \leq 7$  (ou  $\text{lin} \in [1,8], \text{col} \in [1,7]$ );  $\forall \text{lin}, \forall \text{col} \in \mathbb{Z}$
  - Espaço de Estados:  $S = \{ (1,1), (1,2), \dots, (8,7) \}$
  - Estado Inicial:  $s_0 = (2,2)$
  - Estado(s) Final(is):  $G = \{(8,7)\}$
  - Conjunto de Ações:  $A$   
Uma ação  $a \in A$  pode ser descrita na forma:  $\text{ação}(\alpha, \alpha') \leftarrow \beta$   
*onde:  $\alpha$  é o estado atual,  $\alpha'$  é um estado sucessor e  $\beta$  é o conjunto de condições que devem ser obedecidas*
- Exemplo:  
 $\text{oper}(\text{direita}, [\text{lin}, \text{col}], [\text{lin}, \text{col}+1]) \leftarrow \text{col}+1 \leq 7$

# Algoritmos de busca

✓ **Menor custo:**  $f(s)=g(s)$ ; garante encontrar solução ótima

$g(s)$  é o custo acumulado do caminho que vai do estado inicial até o estado corrente

✓ **Melhor estimativa (busca gulosa):**  $f(s)=h(s)$ ; é + rápido que

a busca pelo menor custo, mas não garante encontrar solução ótima

✓ **A\*:**  $f(s)=g(s)+h(s)$ ; é mais rápido que a busca pelo menor custo e mais

lento que a busca pela melhor estimativa; garante encontrar solução ótima

✓ **Dijkstra:** busca pelo menor custo otimizada por meio de podas;

garante encontrar solução ótima (não permite nós repetidos na árvore)

# Menor Custo

BUSCAMENORCUSTO( $\mathcal{A}, s_0, \mathcal{G}$ )

1  $\Gamma \leftarrow \emptyset$

2  $\Sigma \leftarrow \{s_0\}$

3 enquanto  $\Sigma \neq \emptyset$  faça

4      $s \leftarrow \text{removePrimeiro}(\Sigma)$

5     se  $s \in \mathcal{G}$  então devolva *caminho*( $s$ )

6      $\Gamma \leftarrow \Gamma \cup \{s\}$

7      $\text{insereEmOrdem}(\text{sucessores}_G(s, \mathcal{A}) - \Gamma, \Sigma)$

8 devolva fracasso



# Menor Custo

## Passos para construção da árvore:

1. Coloque o estado inicial ( $s_0$ ) na raiz da árvore;
2. Gere os sucessores do estado inicial e coloque-os no nível 1, com seus respectivos custos [  $f(s)=g(s)$  ];
3. A partir daí, gere os sucessores do estado  $s$  de menor valor associado (usando o conjunto de ações  $A$ ), independentemente do seu nível ou ramificação. A busca termina quando o estado  $s$  selecionado para gerar sucessores é estado final, ou seja, quando  $s \in G$ .

# Dijkstra

Dijkstra( $A, s_0, G$ )

1  $\Gamma \leftarrow \emptyset$ ;

2  $\Sigma \leftarrow \{s_0\}$

3 enquanto  $\Sigma \neq \emptyset$  faça

4      $s \leftarrow \text{removePrimeiro}(\Sigma)$

5     se  $s \in G$  então devolva caminho( $s$ )

6      $\Gamma \leftarrow \Gamma \cup \{s\}$

7      $U \leftarrow \text{gere\_sucessores}G(s, \alpha)$

8     Para cada  $u \in U \wedge u \notin \Gamma$

9         se  $u \notin \Sigma$  então  $\text{insereEmOrdem}(u, \Sigma)$

10        senão  $\text{insereEmOrdem}(u, \Sigma)$  se, e somente se,  $u$  é menor que sua  
réplica em  $\Sigma$

11 devolva fracasso

# Dijkstra

## Passos para construção da árvore:

1. Coloque o estado inicial ( $s_0$ ) na raiz da árvore;
2. Gere os sucessores do estado inicial e coloque-os no nível 1, com seus respectivos custos [  $f(s)=g(s)$  ];
3. A partir daí, gere os sucessores do estado  $s$  de menor valor associado (usando o conjunto de ações  $A$ ), independentemente do seu nível ou ramificação, observando as podas de nós repetidos. A busca termina quando o estado  $s$  selecionado para gerar sucessores é estado final, ou seja, quando  $s \in G$ .

# Melhor Estimativa

BUSCAMELHORESTIMATIVA( $\mathcal{A}, s_0, \mathcal{G}$ )

1  $\Gamma \leftarrow \emptyset$

2  $\Sigma \leftarrow \{s_0\}$

3 enquanto  $\Sigma \neq \emptyset$  faça

4      $s \leftarrow \text{removePrimeiro}(\Sigma)$

5     se  $s \in \mathcal{G}$  então devolva  $\text{caminho}(s)$

6      $\Gamma \leftarrow \Gamma \cup \{s\}$

7      $\text{insereEmOrdem}(\text{sucessoresH}(s, \mathcal{A}) - \Gamma, \Sigma)$

8 devolva fracasso

# Melhor Estimativa

## Passos para construção da árvore:

1. Coloque o estado inicial ( $s_0$ ) na raiz da árvore;
2. Gere os sucessores do estado inicial e coloque-os no nível 1, com seus respectivos custos [  $f(s)=h(s)$  ];
3. A partir daí, gere os sucessores do estado  $s$  de menor valor associado (usando o conjunto de ações  $A$ ), independentemente do seu nível ou ramificação. A busca termina quando o estado  $s$  selecionado para gerar sucessores é estado final, ou seja, quando  $s \in G$ .

# A\*

BUSCA A\* ( $\mathcal{A}, s_0, \mathcal{G}$ )

1  $\Gamma \leftarrow \emptyset$

2  $\Sigma \leftarrow \{s_0\}$

3 enquanto  $\Sigma \neq \emptyset$  faça

4      $s \leftarrow \text{removePrimeiro}(\Sigma)$

5     se  $s \in \mathcal{G}$  então devolva  $\text{caminho}(s)$

6      $\Gamma \leftarrow \Gamma \cup \{s\}$

7      $\text{insereEmOrdem}(\text{sucessores}F(s, \mathcal{A}) - \Gamma, \Sigma)$

8 devolva fracasso

# A\*

## Passos para construção da árvore:

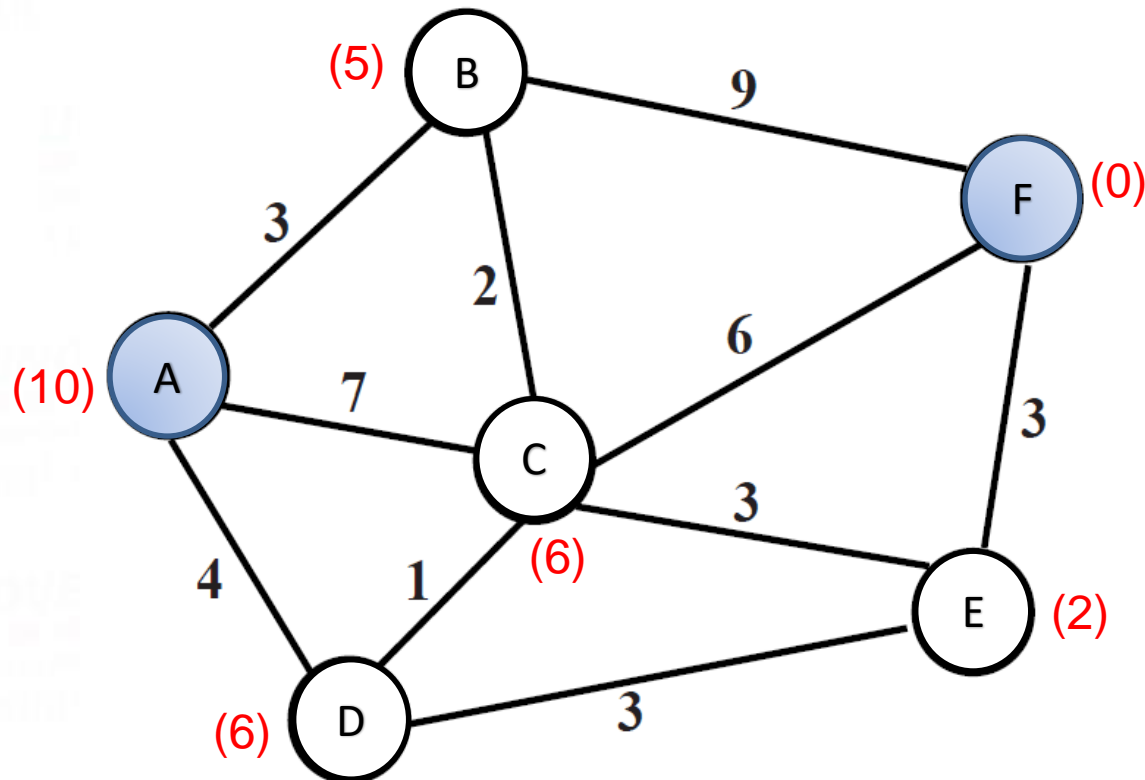
1. Coloque o estado inicial ( $s_0$ ) na raiz da árvore;
2. Gere os sucessores do estado inicial e coloque-os no nível 1, com seus respectivos custos [  $f(s)=g(s)+h(s)$ ];
3. A partir daí, gere os sucessores do estado  $s$  de menor valor associado (usando o conjunto de ações  $A$ ), independentemente do seu nível ou ramificação.  
A busca termina quando o estado  $s$  selecionado para gerar sucessores é estado final, ou seja, quando  $s \in G$ .

# Algoritmos de Busca aplicados ao Problema do Caminho Mínimo - PCM

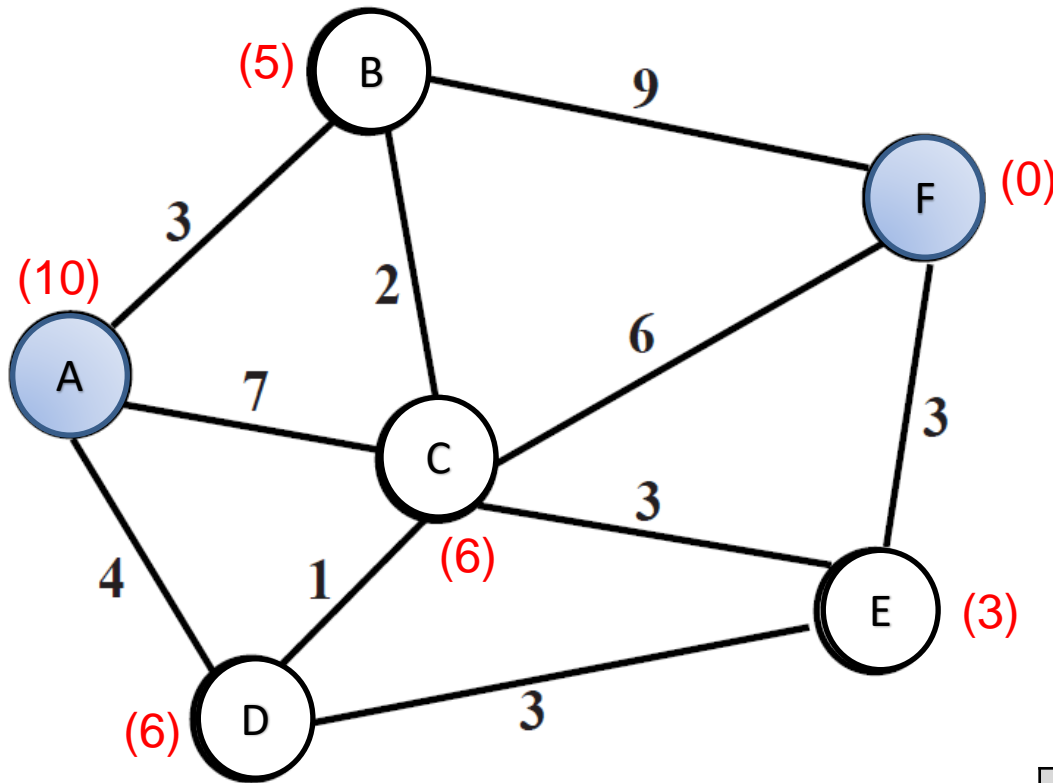


# Exercício 1

Considere o **Problema de Caminho Mínimo** no qual se deseja encontrar um caminho que liga o nó **A** ao nó **F**. No grafo abaixo cada arco indica a distância em km entre os nós (custo) e entre parênteses estão indicadas as estimativas de distâncias até o nó objetivo. Formalize o problema e encontre rotas utilizando as seguintes estratégias de busca: Menor Custo, Dijkstra, Melhor Estimativa e A\*.



# Representação (Exercício 1)



Matriz de Adjacências/Distâncias

	A	B	C	D	E	F
A	$\infty$	3	7	4	$\infty$	$\infty$
B	3	$\infty$	2	$\infty$	$\infty$	9
C	7	2	$\infty$	1	3	6
D	4	$\infty$	1	$\infty$	3	$\infty$
E	$\infty$	$\infty$	3	3	$\infty$	3
F	$\infty$	9	6	$\infty$	3	$\infty$

Matriz heurística

h(A)	h(B)	h(C)	h(D)	h(E)	h(F)
10	5	6	6	3	0

Estado: [c]

$S = \{A, B, C, D, E, F\}$

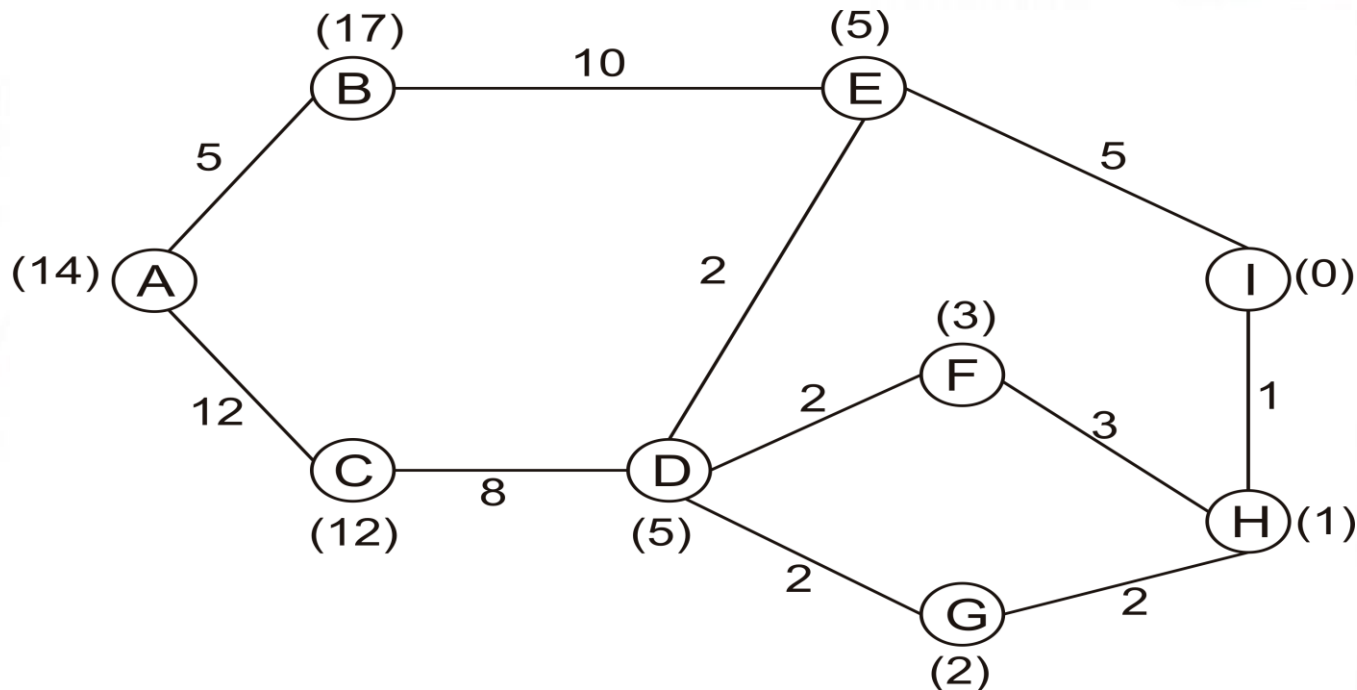
$s_0 = A$

$G = \{F\}$

$A = \{ir(\alpha, \alpha') \leftarrow \text{aresta}(\alpha, \alpha')\}$

# Exercício 2

Considere o **Problema de Caminho Mínimo** no qual se deseja encontrar um caminho que liga o nó **A** ao nó **I**. No grafo abaixo cada arco indica a distância em km entre os roteadores (custo) e entre parênteses estão indicadas as estimativas de distâncias até o nó objetivo. Formalize o problema e encontre rotas utilizando as seguintes estratégias de busca: Menor Custo, Dijkstra, Melhor Estimativa e A\*.



# Referências

ARENALES, Marcos; ARMENTANO, Vinicius; MORABITO, Reinaldo; YANASSE, Horácio; *Pesquisa Operacional para Cursos de Engenharia*; Rio de Janeiro: Editora Campus (Elsevier), 2008. 526p

COELHO, L. C. Série Pesquisa Operacional – Problema do Caixeiro Viajante. Disponível em: <http://www.logisticadescomplicada.com/serie-pesquisa-operacional-%E2%80%93-problema-do-caixeiro-viajante/>  
Acessado em: 28/03/2013.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. Introduction to Algorithms. 2 ed. United States of America: MIT Press, 2001.

LACHTERMACHER, G. Pesquisa Operacional na Tomada de Decisões. 4. ed. São Paulo: Pearson/Prentice Hall, 2009.

RUSSEL, S.; NORVIG, P. Artificial intelligence a modern approach. New Jersey: Prentice Hall, 1995.

TAHA, H. A. Pesquisa Operacional. 8. ed. São Paulo: Pearson, 2008.