



UNIVERSIDADE
VILA VELHA
ESPÍRITO SANTO

Inteligência Artificial

Buscas em Espaços de Estados

Prof. Erlon Pinheiro



Buscas Informadas

- **São buscas que usam conhecimento do problema na solução a ser encontrada.**
- **Também conhecidas como Buscas Heurísticas;**
- **Heurística = Estimativa.**



Algoritmo

- 1. Forme uma fila (com prioridade) de trajetórias parciais. Faça a fila inicial consistir da trajetória de comprimento zero, isto é, apenas a raiz da árvore;**
- 2. Até que a fila esteja vazia ou que o nó destino tenha sido encontrado, determine se a primeira trajetória na fila alcança o nó destino.**
 - 2.1 Se a primeira trajetória alcança o nó destino, não faça nada.**
 - 2.2 Se a primeira trajetória não alcançar o nó destino:**
 - 1. Remova a primeira trajetória da fila;**
 - 2. forme novas trajetórias a partir da trajetória removida pela extensão de um passo;**
 - 3. Acrescente as novas trajetórias à fila;**
 - 4. Ordene a fila crescentemente pela soma de custos acumulados até este nó, com uma estimativa de limite inferior do custo restante, com as trajetórias de menor custo na frente;**
 - 5. Se duas ou mais trajetórias alcançarem um nó comum, apague todas as trajetórias, exceto a que alcança o nó comum com o custo mínimo;**
- 3. Se o nó destino foi encontrado, proclame sucesso; senão proclame fracasso.**



Detalhes - BH

- **A nota é uma estimativa do custo de ir-se da raiz até a meta passando por N e é calculada pela fórmula:**

$$f(N) = g(N) + h(N).$$

- **Nesta fórmula, $g(N)$ é o custo conhecido de ir-se da raiz até o nóduo N. Tal custo pode ser calculado somando-se os custos necessários para atravessar os ramos que ligam N à raiz.**



Detalhes - BH

- **A busca heurística reorganiza as trajetórias da fila de espera de modo que aquelas que terminam em nódulos com menor $f(N)$ venham para frente e, portanto, sejam examinadas antes.**
- **Em PROLOG ou Python ou ETC, esta reorganização pode ser realizada facilmente com auxílio de um método de ordenação.**

O problema - BH

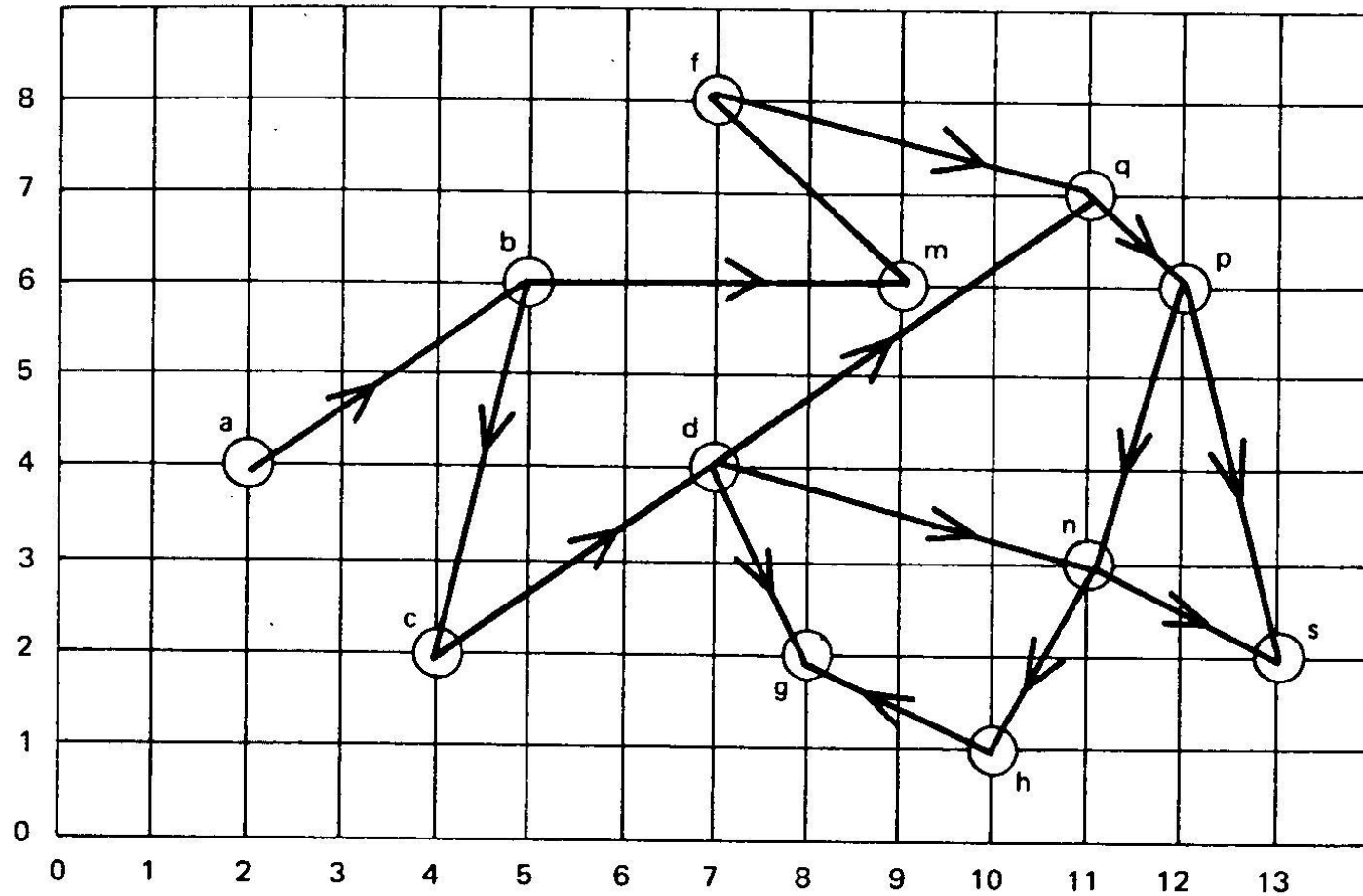


Fig. 5.8.

O problema - BH

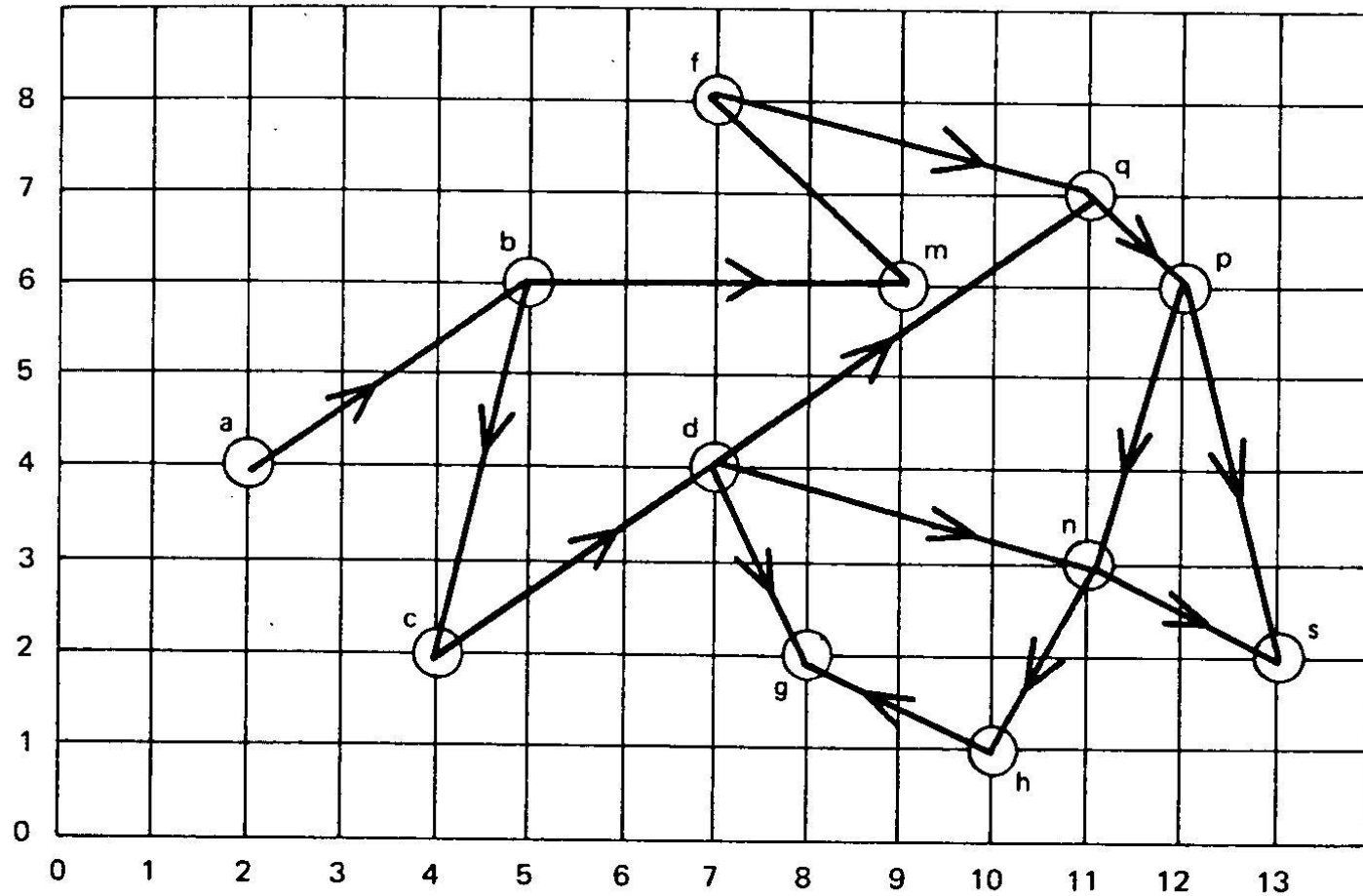


Fig. 5.8.

O problema - BH

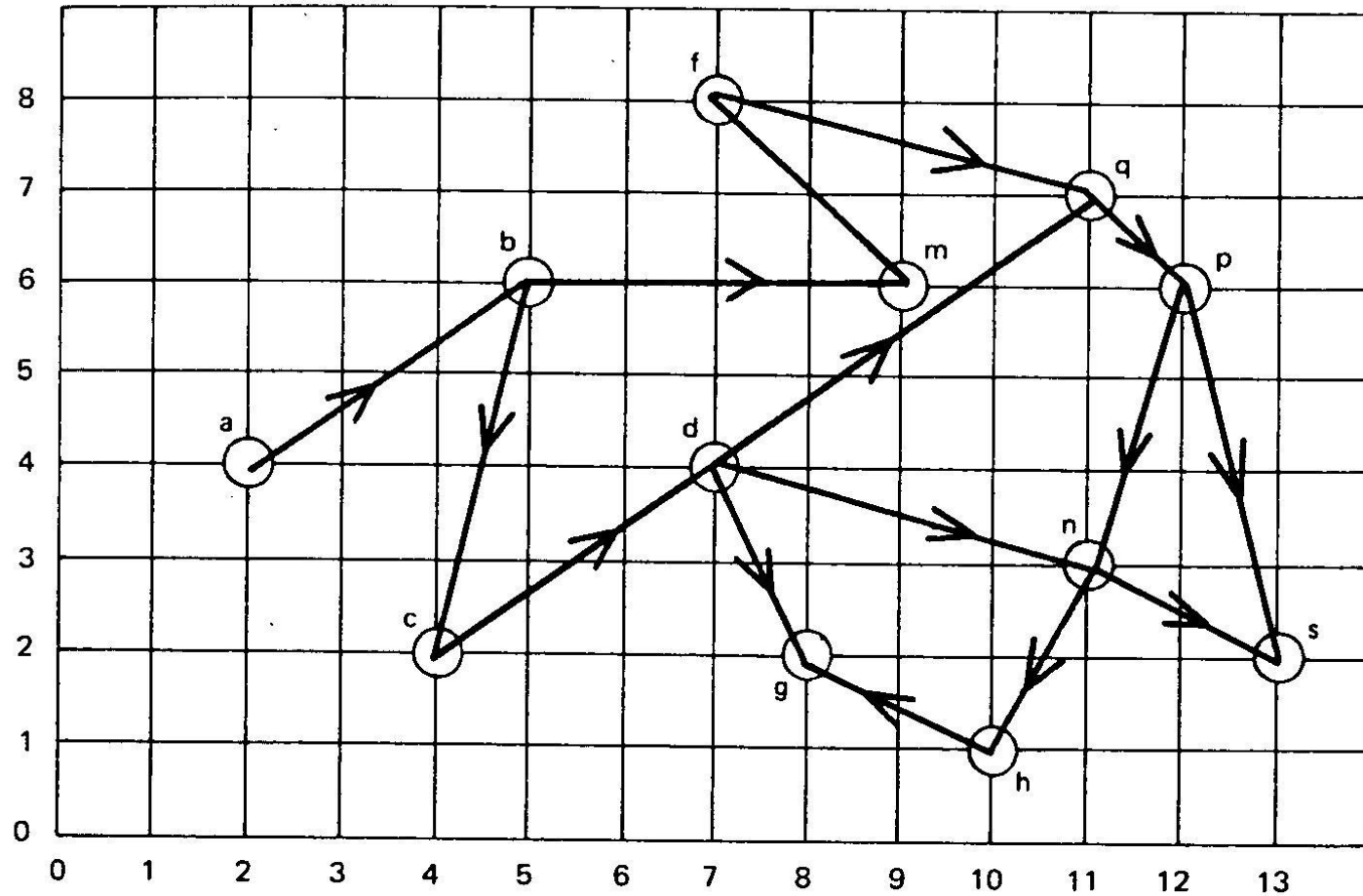


Fig. 5.8.



Detalhes do código - BH

- **Deve-se ter uma representação clara de uma trajetória do mapa;**
- **Fazer os cálculos de $g(N)$, $h(N)$ e $f(N)$.**
- **Usar a estrutura de dados para a fila de prioridades (que pode ser uma fila normal que a cada iteração ocorra uma ordenação) para que a trajetória de menor valor de $f(N)$ fique na frente e, portanto, serão examinadas primeiro.**



Detalhes do código - BH

- Diz-se que um método de busca é admissível se ele sempre encontra uma solução e se esta solução é a de menor custo, isto significa que a busca em largura é admissível;
- Suponha que um programa de busca heurística esteja em um nó N e que por alguma arte de magia seja conhecido exatamente o custo mínimo de ir-se de N até a meta. Este custo é denominado $h^*(N)$.
- Se $h(N)$ for menor ou igual a $h^*(N)$ para qualquer N , o programa que está realizando a busca é admissível;
- Isto é, sempre encontra a trajetória mais curta da raiz à meta. É bom *não confundir encontrar a trajetória mais curta com encontrar mais rapidamente a trajetória.*



Crítica à Busca Heurística

- ***Solução de problemas* usando técnicas de *busca heurística*:**
 - dificuldades em definir e usar a *função de avaliação*
 - não consideram conhecimento genérico do mundo (ou "*senso comum*")
- **Função heurística: compromisso (conflito) entre**
 - tempo gasto na seleção de um nó e
 - redução do espaço de busca
 - Achar o melhor nó a ser expandido a cada passo pode ser tão difícil quanto o problema da busca em geral.



Trabalho – BH

- **Implementar o código da busca heurística para encontrar a menor distância entre as cidades a e s .**
- **O programa deve funcionar tanto para busca heurística quanto para a busca gulosa (quando $f(N) = h(N)$).**



Dicas Finais

- **O Código deve estar bem documentado com as explicações do uso da estrutura de dados utilizada pela busca heurística.**