

Aula 06 - Uso de Heurísticas

Profa. Gabrielly Queiroz

Introdução às Heurísticas

O que são heurísticas?

As heurísticas são estratégias usadas para encontrar soluções rapidamente quando uma busca completa ou exata seria muito demorada. Elas ajudam a reduzir o número de possibilidades analisadas, utilizando estimativas inteligentes para guiar a tomada de decisão.

Exemplos de heurísticas no dia a dia

- Escolher a fila mais curta no supermercado sem calcular exatamente o tempo de espera.
- Usar um atalho no trânsito sem saber com certeza se é a melhor opção.
- Diagnosticar uma doença baseada nos sintomas mais comuns em vez de considerar todas as possibilidades médicas.

A Importância das Heurísticas na IA

Em muitos problemas de IA, **o espaço de busca é gigantesco** e encontrar a solução ótima pode ser impraticável. Heurísticas ajudam a **reduzir o número de opções analisadas**.

As heurísticas são amplamente utilizadas em diversos domínios da IA, como:

- **Algoritmos de busca e otimização:** utilizados em jogos, navegação e problemas de roteamento.
- **Sistemas especialistas:** aplicadas para tomada de decisões baseadas em regras.
- **Aprendizado de Máquina:** influenciam escolhas de features, hiperparâmetros e estratégias de treinamento.
- **Planejamento e raciocínio:** ajudam a definir prioridades e caminhos mais prováveis para atingir um objetivo.

Vantagens e Desvantagens das Heurísticas

Vantagens:

- Encontram soluções rapidamente
- Podem ser ajustadas para diferentes problemas
- Dependem da qualidade da heurística escolhida

Desvantagens:

- Não garantem a melhor solução
- Reduzem o custo computacional
- Podem ser enviesadas ou limitadas

Principais Algoritmos

Busca Sem Informação:

- O algoritmo **não tem conhecimento prévio** sobre onde está o objetivo, então ele simplesmente segue regras fixas para explorar os nós. Explora cegamente.

Busca Com Informação:

- **Usa heurística $h(n)$** → O algoritmo usa uma função que **estima o custo restante até o objetivo** para guiar a busca de maneira mais inteligente.
- **Mais eficiente** → Explora primeiro os caminhos mais promissores, economizando tempo e recursos.
- **Pode encontrar a solução ótima** → Se a heurística for bem escolhida, o algoritmo pode encontrar o caminho mais curto rapidamente.

❖ **Busca Gulosa (Greedy Best-First Search)**

❖ **Algoritmo A***

Função Heurística

A função heurística é uma aproximação do custo para alcançar o objetivo a partir de um determinado estado. Em outras palavras, é uma forma de **estimar o melhor caminho** sem necessariamente explorá-lo por completo.

Matematicamente, podemos definir uma função heurística como:

$h(n)$ = estimativa do custo do nó n até o nó objetivo

Ela **não precisa ser exata**, mas deve ser rápida de calcular e fornecer uma estimativa razoável. Uma boa heurística reduz o número de estados explorados, acelerando a busca.

Busca Gulosa (Greedy Best-First Search)

A **Busca Gulosa** é um algoritmo de busca informada que utiliza apenas a função heurística para decidir qual nó explorar primeiro. Ele escolhe sempre o próximo estado que **parece** estar mais próximo do objetivo, sem considerar o custo real do caminho.

A função usada na Busca Gulosa é:

$$f(n)=h(n)$$

Ou seja, o algoritmo **sempre escolhe o nó com o menor valor de $h(n)$** .

Exemplo

Imagine um problema de navegação onde um agente precisa chegar ao destino. A Busca Gulosa escolheria o próximo ponto baseado **apenas na distância ao destino**, ignorando obstáculos ou caminhos mais longos.

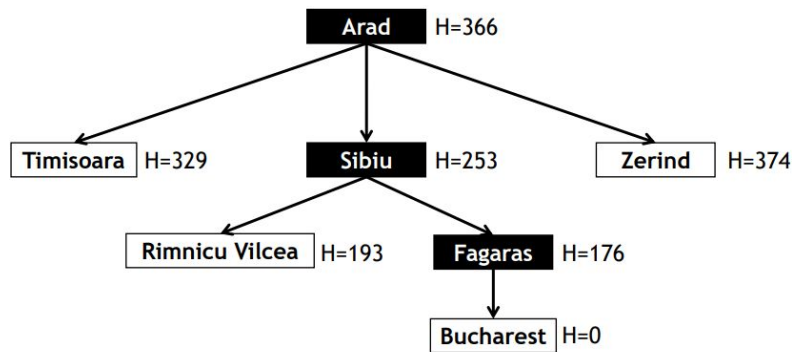
Vantagens da Busca Gulosa:

Simple de implementar.
Pode encontrar soluções rapidamente.

Desvantagens da Busca Gulosa:

Pode ficar presa em mínimos locais, escolhendo um caminho que parece melhor a curto prazo, mas que na verdade é pior.
Não garante encontrar o caminho mais curto (solução ótima).

- **Busca gulosa:** expande o nó com menor valor de $h(x)$.



Algoritmo A* (A-Star)

O **A*** é uma melhoria da Busca Gulosa, combinando a heurística com o custo real do caminho percorrido. Isso resolve o problema da Busca Gulosa de ficar presa em mínimos locais.

A função utilizada pelo A* é:

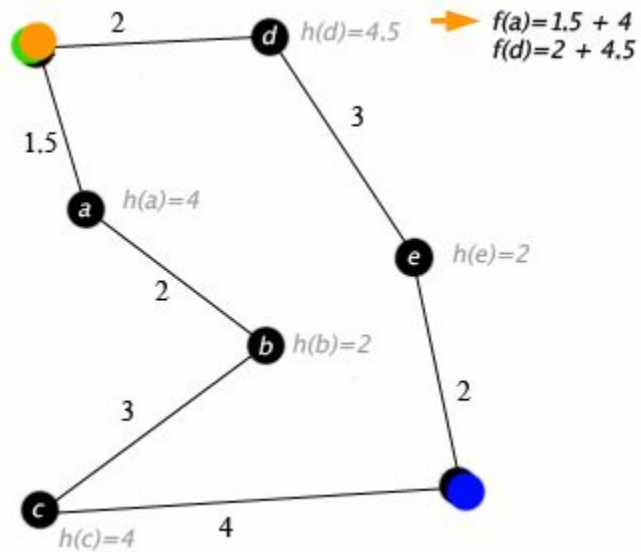
$$f(n)=g(n)+h(n)$$

Onde:

- **g(n)** → Custo real do caminho percorrido até o nó n.
- **h(n)** → Estimativa heurística do custo restante até o objetivo.

O algoritmo A* busca equilibrar **o custo percorrido** com **a previsão do custo restante**, tornando-o um dos algoritmos mais eficientes para busca.

Algoritmo A* (A-Star)



Limitações das Heurísticas

Apesar de úteis, heurísticas **não são perfeitas** e podem levar a erros. Algumas limitações incluem:

1. **Podem ser enviesadas**

- Se uma heurística for baseada em dados ruins, ela pode levar a decisões ruins.
- Exemplo: Se um carro autônomo aprender que uma área tem menos pedestres e, por isso, reduz a atenção, ele pode errar ao não considerar um dia atípico.

2. **Podem não funcionar para todos os problemas**

- Algumas heurísticas funcionam bem em um cenário, mas são ineficazes em outros.

3. **Podem ser manipuladas**

- Algoritmos de recomendação podem ser enganados (exemplo: sistemas de propaganda podem "forçar" recomendações irrelevantes porque seguem heurísticas simples).