

BUSCA INFORMADA

Diferentemente da Busca Exaustiva, onde não se sabe qual o melhor nó de fronteira a ser expandido, a Busca Heurística estima qual o melhor nó da fronteira a ser expandido com base em funções heurísticas, tornando-se assim uma busca não exaustiva do espaço de estados, utilizando atalhos para sua exploração que descartam subespaços sem explorá-los. Em contrapartida, muitas vezes não se tem a garantia de sempre encontrar a melhor solução ou até uma solução em todas as instâncias do problema. Heurísticas tratam de métodos em que, embora a exploração seja feita de forma algorítmica, o progresso é obtido pela avaliação do resultado usando alguma informação contida no problema. Alguns algoritmos de busca heurística global são: Busca gulosa, A*, RBFS, e SMA*.

O algoritmo da busca gulosa pela melhor escolha tem como estratégia expandir o nó supostamente mais próximo a origem (heurística), através de uma estimativa. A distância do nó a expandir ao objetivo é estimada através de uma função heurística. Mas esta estratégia sofre dos mesmos problemas da Busca Exaustiva (cega) em Profundidade, fazendo com que ela não seja completa (pois pode ocorrer loops, por exemplo), não seja ótima (não encontra o menor caminho até o objetivo), e um pior caso de tempo e espaço da ordem de b^m (b = fator de ramificação da árvore, e m = profundidade máxima do espaço de estados).

O algoritmo A* busca expandir o nó com menor valor $\text{Custo}(\text{Caminho})$ na fronteira do espaço de estados, utilizando a heurística $\text{Custo}(\text{Caminho}) = \text{Custo}(\text{Caminho Percorrido}) + \text{Custo}(\text{Caminho Restante})$ ($f(n) = g(n) + h(n)$), o que permite uma simplificação, eliminando caminhos redundantes. Como a exploração é em uma árvore de busca de largura primeira, este algoritmo tem vantagens e limitações parecidas à busca cega de custo uniforme. É uma estratégia completa, quando a heurística é admissível, isto é, quando ela sempre subestima o custo real ($\text{Custo}(\text{Caminho Percorrido})$), para todo nó. Da mesma forma, é ótima para heurísticas admissíveis, garantindo mais eficiência do que qualquer outro algoritmo completo e ótimo usando a mesma heurística, e seu tempo e espaço são no pior caso $O(bd)$, onde d = profundidade da solução de menor custo.