

Sistemas de Inteligentes

Conteúdo

Sistemas de Inteligentes	1
Resolução do 8-Puzzle	3
Descrição do Puzzle:	3
Descrição dos Métodos Implementados:	3
A* com Heurística Manhattan resumo:	3
A* com Heurística Hamming resumo:	3
Pesquisa em Largura (BFS) Resumo:	4
A* com Manhattan	4
A* com Hamming	4
Pesquisa em Largura (BFS)	4
Estudo de Custo de Tempo e Memória	4
Discussão dos Resultados	5
Conclusões Principais	6

Resolução do 8-Puzzle

Descrição do Puzzle:

O 8-Puzzle é um quebra-cabeça deslizante composto por 8 peças numeradas (de 1 a 8) distribuídas num tabuleiro 3x3, onde uma posição é deixada vazia (representada por 0 ou espaço em branco). O objetivo do jogo é transformar uma configuração inicial numa configuração final através de movimentos válidos, deslizando as peças para ocupar a posição vazia. A natureza do puzzle gera um espaço de estados finito, contudo com complexidade combinatória considerável, o que torna a resolução num problema clássico de pesquisa em inteligência artificial.

Descrição dos Métodos Implementados:

O código contempla diferentes abordagens para a resolução do puzzle:

Modo Manual(M): Permite que o jogador mova as peças utilizando as setas do teclado. Cada movimento é executado interactivamente e o número de movimentos é contabilizado.

Modo Automático(A): O jogador pode escolher entre três algoritmos para resolver o puzzle automaticamente.

Modo Comparação(C): Este modo permite ao jogador comparar os diferentes algoritmos e as suas soluções.

A* com Heurística Manhattan resumo:

Utiliza a soma das distâncias horizontais e verticais de cada peça até à sua posição correta na configuração alvo. Esta heurística é admissível e consistente, fornecendo uma boa orientação e, normalmente, tem uma solução ótima com eficiência.

A* com Heurística Hamming resumo:

Calcula o número de peças fora de posição (sem considerar o espaço vazio/0). Embora também seja admissível, esta heurística é menos informativa do que a Manhattan, o que pode levar a uma expansão maior de nós durante a pesquisa o que leva a um maior custo associado.

Pesquisa em Largura (BFS) Resumo:

Explora os estados do puzzle de forma nivelada (por camadas), garantindo encontrar a solução com o menor número de movimentos (solução ótima em termos de custo uniforme). No entanto, BFS pode tornar-se impraticável em termos de tempo e memória para estados com maior profundidade, devido à explosão combinatória do espaço de estados.

A* com Manhattan

Otimização: Garante encontrar a solução ótima, desde que a heurística seja admissível e consistente.

Complexidade: Em geral, a complexidade é exponencial no pior caso, contudo a orientação da heurística Manhattan costuma reduzir significativamente o número de nós expandidos.

A* com Hamming

Otimização: Também encontra a solução ótima, pois a heurística Hamming é admissível.

Complexidade: Pode ser menos eficiente do que a Manhattan, pois a contagem de peças fora do lugar fornece uma estimativa menos refinada do custo restante, aumentando o número de estados explorados.

Pesquisa em Largura (BFS)

Otimização: Garante encontrar a solução com o menor número de movimentos, explora o espaço de estados de forma ordenada por profundidade.

Complexidade: Apresenta uma complexidade de tempo e espaço exponencial; embora seja simples de implementar e garantir a otimização, o uso intensivo de memória torna-o inviável para instâncias mais complexas do puzzle.

Estudo de Custo de Tempo e Memória

BFS: Explora um grande número de estados, implicando um alto custo em tempo de execução e consumo de memória.

A* com Hamming: Expande mais nós do que a Manhattan, tornando-se menos eficiente em puzzles mais complexos.

A* com Manhattan: Garante menor tempo de execução e uso reduzido de memória, menos nós são expandidos até à solução.

Discussão dos Resultados

Eficiência e Otimização: Todos os algoritmos garantem otimização, mas variam em eficiência.

Escalabilidade: A* demonstra melhor escalabilidade em comparação com BFS.

Aplicabilidade: A* com Manhattan é a melhor escolha na maioria dos cenários.

Combinações Testadas e Respostas A seguir, apresentamos cinco exemplos de combinações testadas e as respostas obtidas:

Caso Inicial

```
Configuração inicial:
Introduza 8 números únicos entre 1 e 8 para a configuração inicial (o 0 será inserido automaticamente):
Linha 1: 1 2 3
Linha 2: 4 5 6
Linha 3 (apenas 2 números): 7 8
```

Caso Final

```
Configuração final (objetivo):
Introduza 8 números únicos entre 1 e 8 para a configuração final (o 0 será inserido automaticamente):
Linha 1: 8 7 6
Linha 2: 5 4 3
Linha 3 (apenas 2 números): 2 1
```

Escolha de qual queremos usar

```
Escolha o modo (M para manual, A para automático, C para comparação): a
Algoritmos disponíveis:
1 - A* com Manhattan
2 - A* com Hamming
3 - BFS
Escolha o algoritmo (1, 2 ou 3): 1
```

Solução

```
Solução encontrada com 30 movimentos:
right -> right -> down -> down -> left -> up -> left -> up -> right -> right -> down -> down -> left -> up -> left -> down -> right -> up -> up -> left -> down -> right ->
up -> right -> down -> down -> left -> up -> left -> up
Pressione Enter para começar a solução automática...
```

Solução após a resolução automática.

```
8 7 6
5 4 3
2 1

Movimentos: 30
🎉 Parabéns! Você atingiu a configuração final em 30 movimentos! 🎉
Pressione Enter para o próximo movimento...
```

Comparação das 3 opções

```
Executando A* com Manhattan:
Solução: right -> right -> down -> down -> left -> up -> left -> up -> right -> right -> down -> down -> left -> up -> left -> down -> right -> up -> up -> left -> down ->
right -> up -> right -> down -> down -> left -> up -> left -> up
Número de movimentos: 30

Executando A* com Hamming:
Solução: right -> right -> down -> down -> left -> up -> left -> up -> right -> right -> down -> down -> left -> up -> left -> down -> right -> up -> up -> left -> down ->
right -> up -> right -> down -> down -> left -> up -> left -> up
Número de movimentos: 30

Executando BFS:
Solução: down -> right -> down -> right -> up -> up -> left -> down -> down -> right -> up -> up -> left -> down -> down -> left -> up -> up -> right -> down -> down -> left ->
up -> right -> down -> right -> up -> left -> up -> left
Número de movimentos: 30
```

Conclusões Principais

Versatilidade do Código: Modos interativos e automáticos permitem experiência prática e análise de desempenho.

Importância da Heurística: A heurística Manhattan reduz significativamente o espaço de pesquisa.

Trade-off entre Simplicidade e Desempenho: BFS é mais simples, mas ineficiente para problemas complexos.

Aplicação em Problemas Reais: A seleção de heurísticas adequadas é crucial para o desempenho dos algoritmos de pesquisa em inteligência artificial.