Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal do Norte de Minas Gerais
Campus Montes Claros- Bacharel em Ciência da computação

Relatório do Problema 8-Puzzle e 15-Puzzle da Disciplina Algoritmos em Grafos

Daniel Alves Fonseca Neto

Relatório apresentado como requisito para obtenção de nota na disciplina de Algoritmos em Grafos do curso de Ciência da Computação do Instituto Federal do Norte de Minas Gerais - Campus Montes Claros.

Montes Claros/2024

1. Introdução

O problema abordado é o 8-puzzle, que consiste em um quebra-cabeça deslizante com uma configuração 3x3 onde nove peças são movidas para transformar um estado inicial em um estado objetivo, trocando a posição da peça vazia. A implementação utiliza o algoritmo A* para encontrar a solução ótima de forma eficiente. O A* combina o custo real do caminho percorrido com uma estimativa heurística do custo restante até o objetivo, utilizando a função de avaliação $f(n) = g(n) + h(n)$, onde $g(n)$ é o custo do caminho atual e $h(n)$ é a heurística baseada na distância de Manhattan.

A implementação do A* foi desenvolvida em C++ e inclui funcionalidades para medir o tempo de execução e contar o número de estados avaliados. Além disso, o programa fornece a solução ótima passo a passo, permitindo tanto a execução completa quanto a visualização interativa dos movimentos necessários para resolver o quebra-cabeça. A abordagem também pode ser expandida para o 15-puzzle e o algoritmo IDA*, utilizando o mesmo princípio com a adaptação necessária para o aumento de complexidade.

2. Metodologia

Para o desenvolvimento do sistema de resolução dos puzzles 8-puzzle e 15-puzzle, foram empregadas abordagens distintas devido às diferenças de complexidade entre os dois quebra-cabeças. No 8-puzzle, foi implementado o algoritmo A*, utilizando a função de avaliação $f(n) = g(n) + h(n)$. Neste contexto, $g(n)$ representa o custo do caminho percorrido até o estado atual, e $h(n)$ é a heurística que estima o custo restante até o objetivo. Duas heurísticas foram testadas para o 8-puzzle: a distância de Manhattan pura, que calcula a soma das distâncias absolutas das peças em relação às suas posições alvo, e a distância de Manhattan combinada com o conceito de conflito linear. O conflito linear considera pares de peças que estão fora de lugar e que trocariam de posição se movidas uma vez, oferecendo uma estimativa mais refinada do custo total até o estado objetivo.

No 15-puzzle, devido à maior complexidade e ao espaço de busca expandido, foi utilizado o algoritmo IDA* (Iterative Deepening A*). O IDA* combina a busca em profundidade com uma heurística para controlar o limite de profundidade, que aumenta iterativamente. Assim como no 8-puzzle, as heurísticas testadas foram a distância de Manhattan pura e a combinação de Manhattan com conflito linear. A heurística com conflito linear foi incluída para comparar seu desempenho com a heurística de Manhattan pura.

A implementação do IDA* também incorporou uma abordagem de multithreading para melhorar a eficiência da busca. Foram utilizados até dois threads simultaneamente: o primeiro thread prioriza a expansão de filhos com o maior valor de $f(n)$, enquanto o segundo thread prioriza filhos com o menor valor de $f(n)$. Esse esquema de multithreading foi aplicado para explorar diferentes aspectos do espaço de busca e otimizar o processo de resolução.

A metodologia incluiu a análise do tempo de execução e o número de estados avaliados para cada configuração de heurística e uso de multithreading. Os dados coletados foram utilizados para avaliar a eficiência das heurísticas e do esquema de processamento paralelo, assegurando que as soluções encontradas fossem otimizadas conforme os critérios estabelecidos.

3. Resultados

Após a implementação dos Algoritmos, foram realizados alguns testes de execução.

3.1. 8-Puzzle

Para o 8-Puzzle foi definido algumas entradas

Num	Estado Inicial	Estado Final	Solução Ótima
1	5 6 2 7 1 8 3 4 0	0 1 2 3 4 5 6 7 8	22
2	5 3 2 8 7 1 4 0 6	0 1 2 3 4 5 6 7 8	23
3	1 0 7 5 2 4 3 8 6	0 1 2 3 4 5 6 7 8	17
4	3 6 4 2 8 0 5 7 1	0 1 2 3 4 5 6 7 8	25
5	0 6 1 7 4 2 3 8 5	0 1 2 3 4 5 6 7 8	16
6	0 3 8 2 1 4 7 6 5	0 1 2 3 4 5 6 7 8	22
7	0 8 2 1 5 6 7 3 4	0 1 2 3 4 5 6 7 8	22
8	5 0 2 6 4 8 1 7 3	0 1 2 3 4 5 6 7 8	21
9	3 0 8 4 1 2 6 7 5	0 1 2 3 4 5 6 7 8	17
10	8 6 7 2 5 4 3 0 1	1 2 3 4 5 6 7 8 0	31

3.1.1.

8-puzzle utilizando o algoritmo A*, foram testadas duas configurações heurísticas. A configuração "MD" refere-se à aplicação do A* com a heurística da distância de Manhattan. A configuração "MD + CL" refere-se à utilização da heurística da distância de Manhattan combinada com a heurística de conflito linear.

3.1.1.1. Configuração "MD"

Heurística MD			
Num	**Movimentos**	**Nós Visitados**	**T. Execucao (ms)**
1	22	1404	1
2	23	1631	1
3	17	118	0
4	25	2128	2
5	16	142	0
6	22	1185	1
7	22	616	0
8	21	593	0
9	17	195	0
10	31	12589	18

3.1.1.2. Configuração "MD + CL"

Heurística MD + CL			
Num	**Movimentos**	**Nós Visitados**	**T. Execucao (ms)**
1	22	627	1
2	23	647	0
3	17	74	0
4	25	690	1
5	16	110	0
6	22	661	1
7	22	496	1
8	21	294	0
9	17	162	0

10	31	5706	9

3.1.2. 8-puzzle utilizando o algoritmo IDA*, foram testadas duas configurações heurísticas. A configuração "MD" refere-se à aplicação do IDA* com a heurística da distância de Manhattan. A configuração "MD + CL" refere-se à utilização da heurística da distância de Manhattan combinada com a heurística de conflito linear.

3.1.2.1. Configuração "MD"

Heurística MD			
Num	**Movimentos**	**Nós Visitados**	**T. Execucao (ms)**
1	22	2099	1
2	23	1135	0
3	17	127	0
4	25	1737	0
5	16	115	0
6	22	1741	0
7	22	1171	0
8	21	629	0
9	17	577	0
10	31	13609	7

3.1.2.2. Configuração "MD + CL"

Heurística MD + CL			
Num	**Movimentos**	**Nós Visitados**	**T. Execucao (ms)**
1	22	1195	1
2	23	407	0
3	17	61	0
4	25	771	0
5	16	65	0

6	22	771	0
7	22	606	0
8	21	385	0
9	17	161	0
10	31	7722	7

3.2. 15-Puzzle

Para o 15-Puzzle foi definido algumas entradas

Num	Estado Inicial	Estado Final	Solução Ótima
1	15 2 12 11 14 13 9 5 1 3 8 7 0 10 6 4	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	65
2	14 13 15 7 11 12 9 5 6 0 2 1 4 8 10 3	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	57
3	13 5 4 10 9 12 8 14 2 3 7 1 0 15 11 6	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	55
4	14 7 8 2 13 11 10 4 9 12 5 0 3 6 1 15	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	59
5	5 12 10 7 15 11 14 0 8 2 1 13 3 4 9 6	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	56
6	4 7 14 13 10 3 9 12 11 5 6 15 1 2 8 0	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	56
7	14 7 1 9 12 3 6 15 8 11 2 5 10 0 4 13	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	52
8	2 11 15 5 13 4 6 7 12 8 10 1 9 3 14 0	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	52
9	12 11 15 3 8 0 4 2 6 13 9 5 14 1 10 7	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	50
10	3 14 9 11 5 4 8 2 13 12 6 7 10 1 15 0	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	46

3.2.1. 8-puzzle utilizando o algoritmo IDA*, foram testadas três configurações heurísticas. A configuração "MD" refere-se à aplicação do IDA* com a heurística da distância de Manhattan. A configuração "MD + CL" refere-se à utilização da heurística da distância de Manhattan combinada com a heurística de conflito linear. A configuração “Threads” que se refere ao “MD+CL” com uso de threads.

3.2.1.1. Configuração "MD"

Heurística MD			
Num	**Movimentos**	**Nós Visitados**	**T. Execucao (s)**
1	65	-957062074	2456
2	57	174122539	124
3	55	13543132	9
4	59	285668347	191
5	56	56316560	35
6	56	6243604	4
7	52	3302195	2
8	52	84062905	57
9	50	16925819	11
10	46	1611935	0

3.2.1.2. Configuração "MD + CL"

Heurística MD + CL			
Num	**Movimentos**	**Nós Visitados**	**T. Execucao (s)**
1	65	197932218	480
2	57	3816820	10
3	55	2905209	8
4	59	69853115	199
5	56	6133397	17
6	56	1431161	4
7	52	471058	0
8	52	20272139	58
9	50	934396	2
10	46	99169	0

3.2.1.3. Configuração "Threads"

Heurística Threads			
Num	**Movimentos**	**Nós Visitados**	**T. Execucao (s)**
1	65	197932218	480
2	57	3816820	11
3	55	2583551	7
4	59	57071936	180
5	56	3444086	10
6	56	1431161	4
7	52	471058	0
8	52	20272138	62
9	50	934396	2
10	46	36582	0

4. Conclusão

No caso do 8-puzzle utilizando o algoritmo A*, a heurística da distância de Manhattan demonstrou um bom desempenho na resolução dos quebra-cabeças. No entanto, apesar da eficiência na determinação da solução, o número de nós acessados durante a busca foi relativamente alto. Em contraste, ao utilizar a heurística que combina a distância de Manhattan com o conflito linear, observou-se uma melhoria significativa no desempenho, com uma média de aproximadamente metade do número de acessos aos nós em comparação com a heurística de Manhattan pura.

Para o 8-puzzle com o algoritmo IDA*, tanto a heurística de Manhattan quanto a heurística combinada com o conflito linear resultaram em um número de acessos aos nós levemente superior ao observado com o A*. No entanto, apesar do aumento no número de nós acessados, o tempo total de execução foi mais eficiente com a heurística IDA* em comparação ao A*. Esta diferença, embora pequena, destaca a vantagem do IDA* em termos de eficiência temporal, mesmo quando o número de nós acessados é um pouco maior.

No caso do 15-puzzle utilizando o algoritmo IDA* com a heurística da distância de Manhattan, o número elevado de acessos aos nós resultou em um tempo considerável para encontrar a solução ótima. Este elevado custo computacional é refletido na dificuldade do algoritmo em explorar eficientemente o espaço de busca. Por outro lado, ao empregar a heurística que combina a distância de Manhattan com o conflito linear, observou-se uma melhora significativa tanto no número de nós acessados quanto no tempo necessário para alcançar a solução.

Além disso, ao testar a heurística combinada com o conflito linear em um ambiente multithread, foi notado que o uso de threads resultou em uma melhoria significativa em determinados casos. A implementação de até dois threads mostrou uma redução considerável no tempo de execução e no número de acessos aos nós, destacando a eficácia do processamento paralelo em otimizar a performance do algoritmo IDA*.