INSTITUTO FEDERAL DO ESPÍRITO SANTO – CAMPUS SERRA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

CLÉBER DE JESUS SALUSTIANO (20202BSI0268)

Trabalho 1: Solução de Problemas por Busca

Trabalho solicitado pelo professor
de Inteligência Artificial,
Sérgio Nery Simões

SERRA – ES
2023

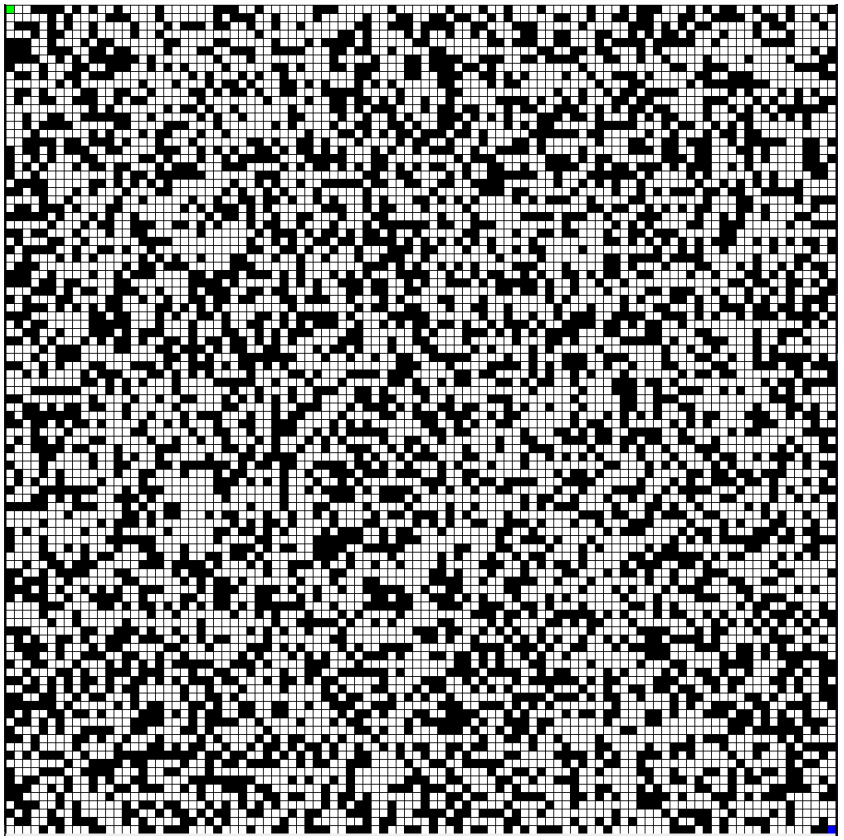
1. INTRODUÇÃO **3**
2. FUNDAMENTAÇÃO TEÓRICA **4**
2.1. DEPTH-FIRST SEARCH (DFS) 4
2.2. A* SEARCH 4
2.3. UNIFORM COST SEARCH (UCS) 4
3. EXPERIMENTOS **5**
3.1. CONFIGURAÇÃO DO LABIRINTO 5
3.2. EXECUÇÕES MÚLTIPLAS 5
3.3. MÉTRICAS AVALIADAS 5
3.4. RESULTADOS ESPERADOS 6
3.5. MÁQUINA UTILIZADA 6
4. RESULTADOS **6**
5. REFERÊNCIAS **9**

1. INTRODUÇÃO

A otimização do processo de busca de caminhos é uma tarefa fundamental em diversas áreas da ciência da computação, da robótica à inteligência artificial. Algoritmos de busca desempenham um papel crucial na resolução de problemas que envolvem a navegação de um ponto de origem para um destino em ambientes complexos. Nesse contexto, a escolha do algoritmo de busca apropriado desempenha um papel determinante na eficiência e eficácia da resolução desses problemas.

Este relatório aborda a avaliação e comparação de três algoritmos de busca amplamente usados - Depth-First Search (DFS), A* e Uniform Cost Search (UCS) - em um labirinto de dimensões 100x100. O principal objetivo deste estudo é analisar o desempenho e a eficácia desses algoritmos em ambientes complexos e dimensionados, com foco em métricas cruciais, como custo total do caminho, número de passos, nós expandidos e tempo de execução. Para garantir a consistência das conclusões, cada algoritmo será testado repetidamente no mesmo labirinto, com foco na obtenção da média do tempo de execução.

Figura 1 - Labirinto 100x100 utilizado durante a utilização dos algoritmos, sendo o ponto verde o ponto inicial e o ponto azul o ponto objetivo.



Fonte: elaborada pelo autor.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. DEPTH-FIRST SEARCH (DFS)

O Depth-First Search é um algoritmo de busca não informada que prioriza a exploração em profundidade. Ele começa pela raiz e avança para um nó filho, indo o mais fundo possível antes de retroceder para explorar outros ramos. Isso significa que o DFS pode explorar um único caminho até o fim antes de considerar outras opções, o que pode ser útil em problemas onde a solução é encontrada em profundidade.

No entanto, o DFS pode não ser adequado para problemas em que é importante encontrar a solução mais próxima do nó de origem, uma vez que ele não considera custos ou distâncias. Também é suscetível a entrar em loops infinitos em grafos com ciclos, a menos que seja implementada uma verificação para nós visitados.

2.2. A* SEARCH

O A* Search é um algoritmo de busca informada que combina a busca de custo uniforme com uma heurística. A heurística é uma estimativa do custo restante para atingir o objetivo a partir de um determinado nó. O A* prioriza a exploração de nós que têm baixo custo atual (a partir do nó de origem) e baixo custo estimado (de acordo com a heurística).

O A* é garantido para encontrar a solução ótima, desde que a heurística seja admissível (nunca superestima o custo real) e o grafo não contenha ciclos com custos negativos. O A* é eficiente em encontrar caminhos curtos em grafos com muitos nós, desde que a heurística seja bem escolhida, pois ele evita a expansão de caminhos que parecem improváveis de levar à solução ótima.

2.3. UNIFORM COST SEARCH (UCS)

O Uniform Cost Search é um algoritmo de busca que prioriza a exploração dos caminhos com o menor custo acumulado. Ele é especialmente útil para encontrar o caminho mais curto em grafos ponderados. O UCS usa uma fila de prioridade para garantir que os caminhos mais baratos sejam explorados primeiro, considerando o custo acumulado desde o nó de origem.

O UCS é garantido para encontrar a solução ótima, desde que o custo das arestas seja não negativo. No entanto, ele pode ser lento em grafos com custos elevados, pois explora muitos caminhos antes de encontrar a solução.

Em resumo, o DFS é adequado para exploração em profundidade, o UCS prioriza o menor custo e o A* combina custos e heurísticas para encontrar a solução mais eficaz e ótima quando aplicável. Cada um desses algoritmos tem suas vantagens e limitações, e a escolha depende das características do problema em questão.

3. EXPERIMENTOS

O experimento proposto visa comparar três algoritmos de busca em grafos, nomeadamente o Depth-First Search (DFS), o algoritmo A* e o Uniform Cost Search (USC), quando aplicados para encontrar um caminho do ponto A ao ponto B em um labirinto. Os principais objetivos deste experimento são avaliar o desempenho desses algoritmos em termos de diversas métricas, incluindo "Custo total do caminho", "Número de passos", "Número total de nós expandidos" e "tempo de execução". Aqui está uma descrição mais detalhada do experimento:

3.1. CONFIGURAÇÃO DO LABIRINTO

O labirinto é modelado como um grafo, onde os nós representam diferentes posições no labirinto, e as arestas representam as conexões entre essas posições. O ponto A é a posição de partida, e o ponto B é a posição de destino. O labirinto é desenhado de tal forma que contenha obstáculos e caminhos possíveis. Cada algoritmo usará o mesmo labirinto 100x100 em todas as execuções do experimento.

3.2. EXECUÇÕES MÚLTIPLAS

O experimento será repetido 1000 vezes com os três algoritmos, utilizando o mesmo labirinto, a fim de mitigar a influência de elementos em segundo plano no tempo de execução. Isso permitirá a avaliação do desempenho em várias configurações do labirinto e a compreensão de como os algoritmos se comportam em diferentes iterações, garantindo que os resultados sejam representativos em relação ao tempo de execução.

3.3. MÉTRICAS AVALIADAS

Custo Total do Caminho: Essa métrica avalia o custo total do caminho encontrado pelos algoritmos do ponto inicial ao ponto final no labirinto. O custo é determinado somando os custos das arestas percorridas.

Número de Passos: O número de passos é a contagem do número de nós visitados ou expandidos pelos algoritmos para alcançar o ponto final a partir do ponto de partida.

Número Total de Nós Expandidos: Essa métrica avalia quantos nós no grafo foram expandidos durante a busca. Isso pode dar uma ideia da eficiência computacional dos algoritmos.

Tempo de Execução: O tempo de execução mede quanto tempo cada algoritmo leva para encontrar o caminho do ponto inicial ao ponto final. Isso ajuda a avaliar o desempenho em termos de eficiência temporal.

3.4. RESULTADOS ESPERADOS

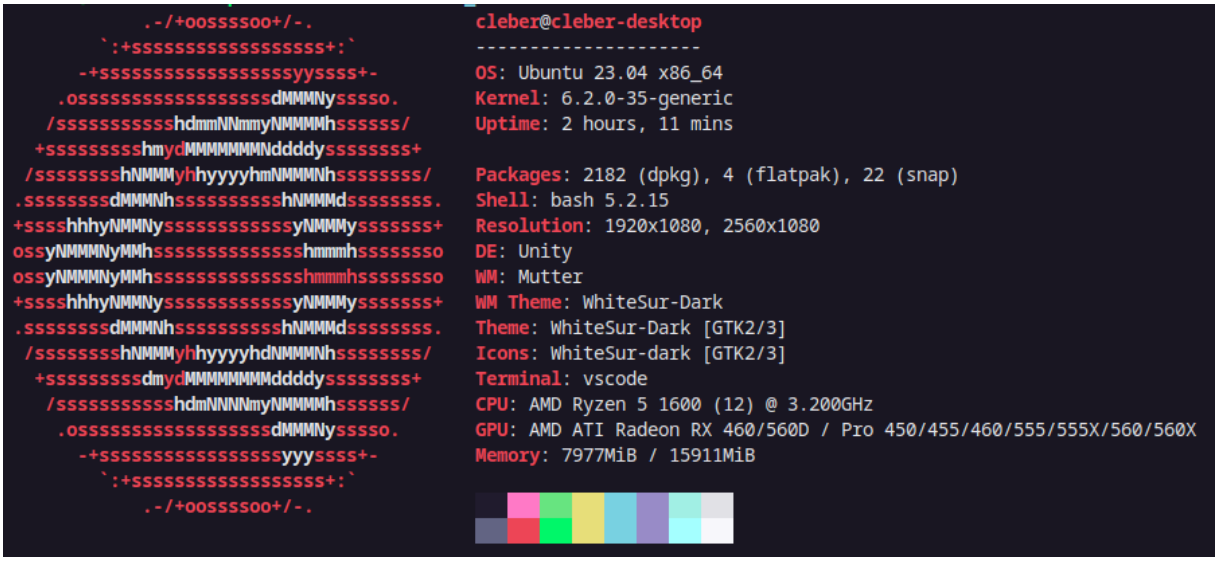
Espera-se que o algoritmo DFS seja o mais rápido em termos de tempo de execução, mas pode não encontrar o caminho mais otimizado. Por outro lado, o algoritmo A* deve encontrar um caminho otimizado, mas pode ser mais lento devido ao cálculo da função heurística. O USC é esperado para encontrar o caminho mais otimizado garantidamente, mas a sua eficiência pode variar dependendo da complexidade do labirinto.

O experimento será executado 100 vezes com os três algoritmos, usando o mesmo labirinto, para garantir a robustez das conclusões e avaliar o desempenho em várias configurações do labirinto. Isso ajudará a avaliar como os algoritmos se comportam em diferentes iterações e garantirá que os resultados sejam representativos.

3.5. MÁQUINA UTILIZADA

A máquina utilizada possui a especificação descrita na Figura 2.

Figura 2 - Configuração do computador utilizado para rodar os algoritmos.



Fonte: elaborada pelo autor.

4. RESULTADOS

Na Tabela 1 temos os resultados das métricas de cada algoritmo executado, baseado em cada métrica descrita anteriormente. Os resultados são coerentes e conforme o esperado, pois o DFS é o mais rápido de todos os algoritmos, mas encontrou o caminho com maior número de passos, maior custo e nós total expandidos, isso se deve ao fato de que o DFS não busca o caminho com menor custo, nem com o menor número de passos e busca o caminho mais profundo possível.

O A* possui um tempo de execução perto do DFS porém maior, mas o ganho nos demais aspectos são extremamente grandes, onde o custo total foi bem menor, o número de passos também além da quantidade de nós expandidos serem menor também, visto que ele faz uma busca informada, o que o DFS não faz.

O UCS possui o maior tempo de execução de todos, porém possui o menor custo total do caminho e o menor número de passos até o objetivo. Mas para isso ele expandiu o máximo de nós possível e assim tem o maior número de nós expandidos dentre os algoritmos, com mais de 6000 nós expandidos enquanto os outros estão na faixa de 300.

Tabela 1 - Resultado dos algoritmos

	Custo total do caminho	Número de passos	Número total de nós expandidos	Tempo de execução (segundos)
DFS	327,10	260	355	0.01352
A*	170.10	132	317	0.01424
UCS	164.35	125	6058	2.29753

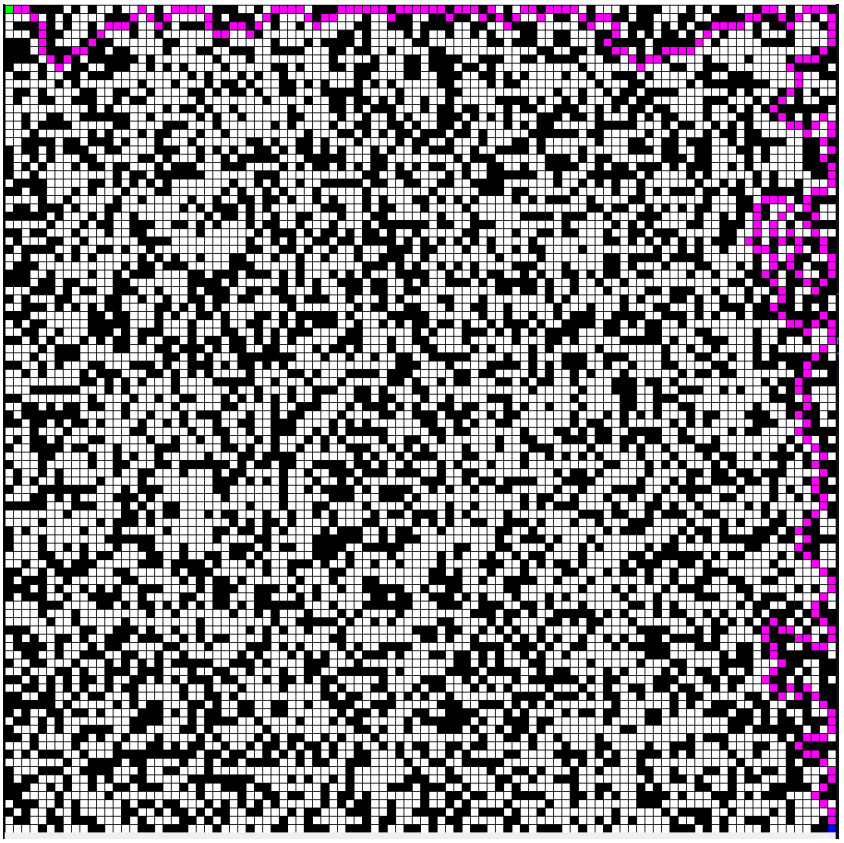
Fonte: elaborada pelo autor.

Com isso, podemos observar que o resultado esperado na teoria é corretamente refletido pelos resultados observados na prática. Cada algoritmo possui sua especificidade e depende da aplicação. Se for necessário encontrar um caminho de maneira mais rápida, sem se preocupar com os custos ou o número de passos, e não for relevante saber a localização do ponto de destino, e a implementação for de baixa complexidade, o DFS (Figura 3) atenderá à necessidade.

Se for necessário encontrar um caminho em que a busca seja informada, levando em consideração o custo total do percurso, com poucos passos, mas com tempo de execução reduzido, o algoritmo A* (Figura 4) será a escolha adequada.

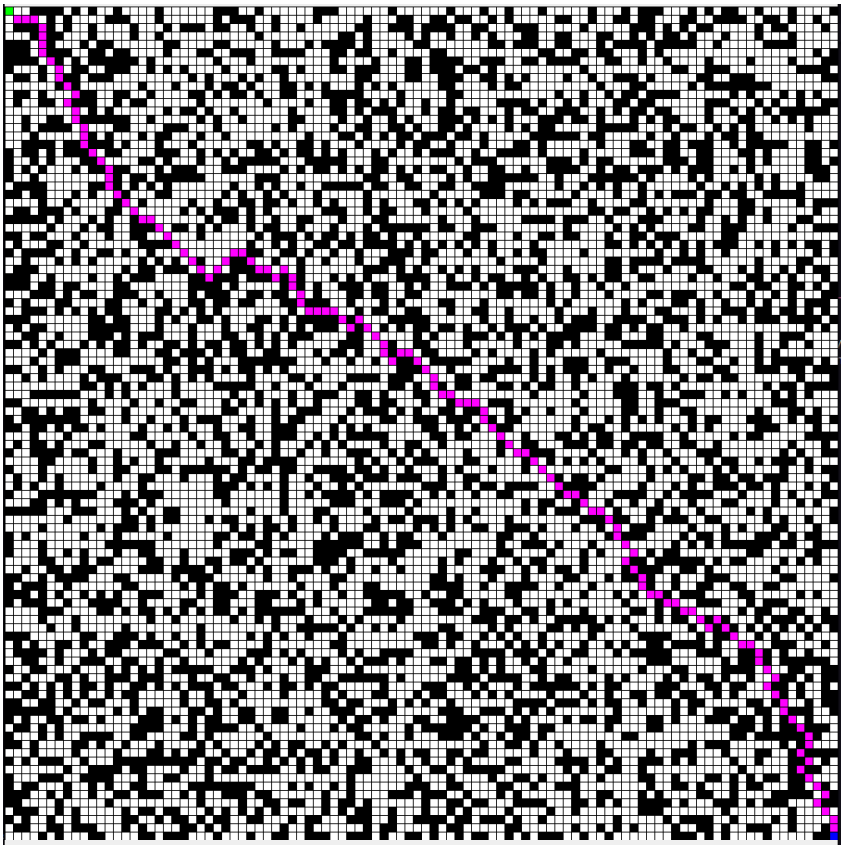
No entanto, se a prioridade for encontrar o melhor caminho, ou seja, a combinação mais eficaz entre o menor número de passos e o menor custo total, sem se preocupar com o tempo de execução, o UCS (Figura 5) será a opção ideal.

Figura 3 - Caminho percorrido usando o algoritmo DFS



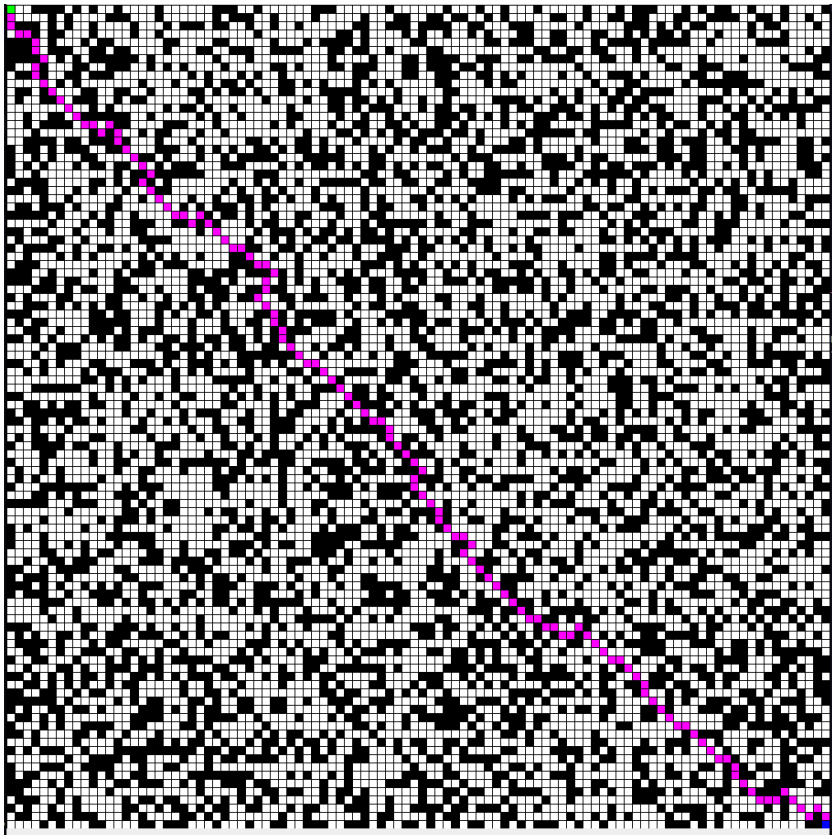
Fonte: elaborado pelo autor

Figura 4 - Caminho percorrido usando o algoritmo A*



Fonte: elaborado pelo autor

Figura 5 - Caminho percorrido usando o algoritmo UCS



Fonte: elaborada pelo autor.

5. REFERÊNCIAS

Russell, Stuart J. 1962-, et al. Artificial Intelligence: A Modern Approach. 3rd ed. Upper Saddle River, NJ, Prentice Hall, 2010.

DEPTH-FIRST SEARCH. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2020. Disponível em: <https://en.wikipedia.org/w/index.php?title=Depth-first_search&oldid=1146291410>. Acesso em: 12 set. 2020.

A* SEARCH ALGORITHM. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2023. Disponível em: <https://en.wikipedia.org/w/index.php?title=A*_search_algorithm&oldid=1179925908>. Acesso em: 24 set. 2023.

DIJKSTRA'S ALGORITHM. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2023. Disponível em: <https://en.wikipedia.org/w/index.php?title=Dijkstra%27s_algorithm&oldid=1180972777>. Acesso em: 17 out. 2023.