Universidade Federal de Minas Gerais

DCC642 - Introdução à Inteligência Artificial (2025/2)
TP1: Busca no Espaço de Estados

Raphael Henrique Braga Leivas - 2020028101

1 Introdução

Neste trabalho, algoritmos de busca são implementados em Python para encontrar o caminho de menor custo em um mapa com estrutura de grade. Os algoritmos são comparados em diferentes mapas para verificar experimentalmente as diferenças entre eles.

2 Objetivos

Os objetivos principais do trabalho são:

- Implementar em Python os algoritmos de busca em largura (BFS), busca em profundidade (DFS), busca de custo uniforme (UCS), busca gulosa por melhor escolha (Greedy) e busca A*, com heurísticas euclidiana e de Manhattan;
- Comparar os algoritmos em 7 casos de teste com diferentes mapas, comparando não só os caminhos retornados, mas também o número de nós expandidos e o custo do caminho final.

3 Casos de Teste

Nesta seção são abordados 7 casos de teste para evidenciar as principais diferenças entre os algoritmos estudados.

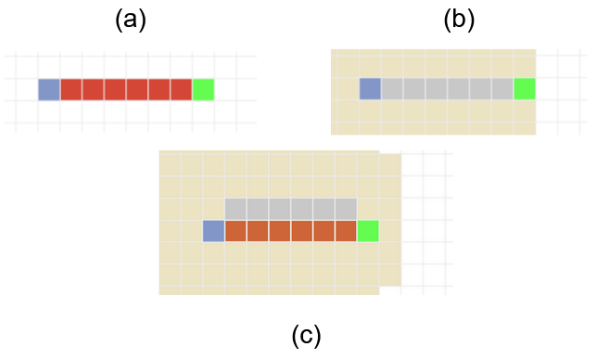
3.1 BFS não é ótimo

O BFS não garante otimalidade se os custos forem diferentes, como mostra a Figura 1.

Na Figura 1 (a), temos que o caminho direto do início ao alvo passa por casas de custo elevado (9), enquanto as demais casas possuem custo unitário. O BFS, por explorar os nós em camadas, acaba encontrando o caminho da Figura 1(b) que vai diretamente até o alvo, passando pelos nós custosos e retornando um caminho de custo total 55.

Contudo, claramente existe um caminho melhor que passa pelos nós de custo unitário. Assim, usando o UCS é possível encontrar o caminho da Figura 1 (c), com custo total 8, que é a solução ótima para o problema.

Figura 1: Caso em que o BFS não é ótimo. (a) Mapa inicial, (b) caminho retornado pelo BFS, (c) caminho ótimo encontrado pelo UCS.



3.2 BFS é equivalente à UCS

BFS e UCS são equivalentes quando todos os custos são iguais (Russell and Norvig 2020), como mostra a Figura 2 (a). Todas as arestas do mapa possuem custo unitário, de modo que a solução encontrada pelo BFS na Figura 2 (b) é igual à solução do UCS na Figura 2 (c). Além disso, note que a maneira em que os nós são explorados é bastante parecida, avançando radialmente a partir do nó inicial.

3.3 DFS retorna a solução ótima

O DFS em geral não é ótimo uma vez que ele retorna o primeiro caminho encontrado e não necessariamente o de menor custo. Contudo, em alguns casos o DFS pode retornar a solução ótima, como mostra a Figura 3 (a), em que todos os custos são iguais e todas as soluções possíveis possuem a mesma profundidade.

Note que esse caso é bastante específico e não é possível garantir que o DFS sempre retorne a solução ótima.

3.4 Greedy é ótimo

A busca gulosa pela melhor escolha (Greedy) é ótima se todos os custos forem iguais. A Figura 4 (a) mostra um mapa em que todos os custos são unitários, de modo que o Greedy retorna o caminho da Figura 4 (b), que é ótimo.

Figura 2: Caso em que o BFS e o UCS são equivalentes. (a) Mapa inicial, (b) caminho retornado pelo BFS, (c) caminho retornado pelo UCS.

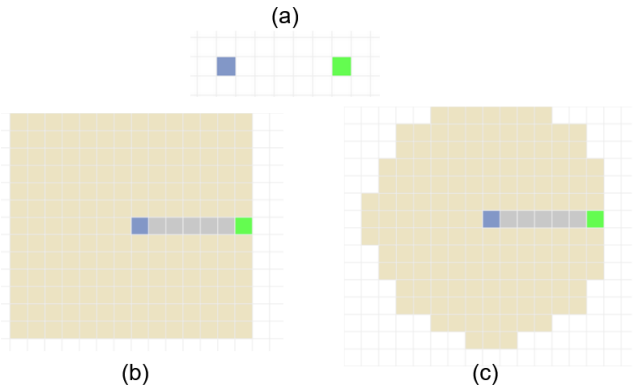


Figura 3: Caso em que o DFS garante otimalidade. (a) Mapa inicial, (b) caminho retornado pelo DFS.

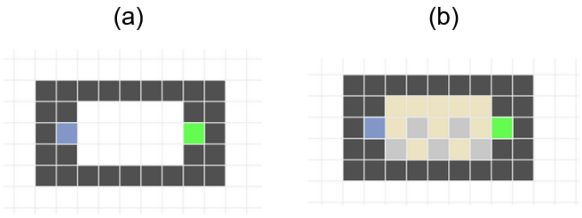


Figura 4: Caso em que o Greedy garante otimalidade (heurística euclidiana). (a) Mapa inicial, (b) caminho retornado pelo Greedy.

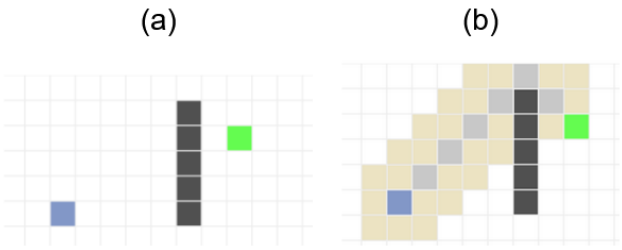
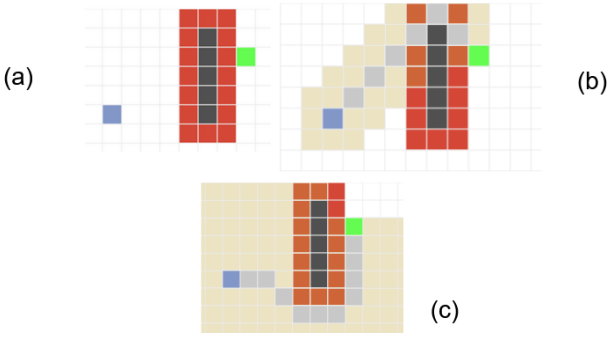


Figura 5: Caso em que o Greedy não garante otimalidade (heurística euclidiana). (a) Mapa inicial, (b) caminho retornado pelo Greedy e (c) caminho ótimo retornado pelo UCS.



3.5 Greedy não é ótimo

No entanto, ao contrário do visto na Seção 3.4, se os custos forem diferentes o Greedy pode não ser ótimo, como mostra a Figura 5 (a). De forma similar ao visto com o BFS na Figura 1, o Greedy acaba escolhendo o caminho que vai direto para o alvo, passando por nós de custo elevado (9) e retornando um caminho de custo total 43.48, sendo que o UCS na Figura 5 (c) encontra o caminho ótimo, com custo total 12. Isso ocorre pois o Greedy está considerando apenas o valor da heurística ao decidir qual nó expandir, ignorando o custo do caminho até aquele nó.

3.6 A* é melhor que UCS

Como o A* utiliza uma heurística para guiar a busca, ele pode ser mais eficiente que o UCS, como mostra a Figura 6 (a). Nesse caso, o A* retorna o caminho da Figura 6 (b) expandindo 910 nós enquato o UCS retorna o caminho da Figura 6 (c) expandindo 1146 nós.

Isso ocorre uma vez que a heuirística usada pelo A* direciona a busca, expandindo menos nós durante o processo. Esse fato é evidenciado observando-se a expansão dos nós visitados na Figura 6, na qual o A* avança mais diretamente em direção ao alvo, enquanto o UCS expande nós que estão muito distantes do nó original que não são vistos pelo A*.

Por fim, note que ambos ainda encontram a mesma solução ótima, com custo total 47.04 e tamanho 41.

3.7 A* é equivalente à UCS

O A* considera o custo total dado pelo soma

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

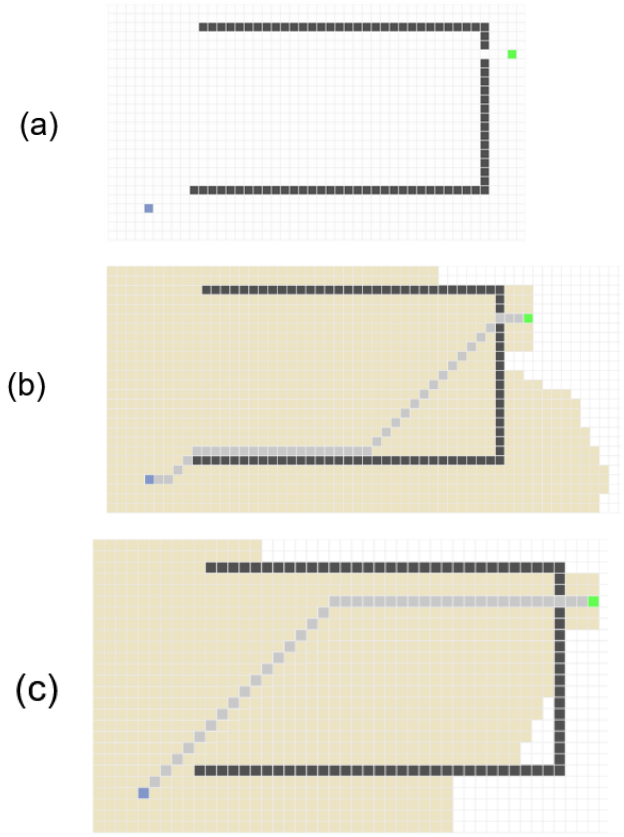
onde:

- $f(n)$: custo total considerado pelo A* no nó n;
- $g(n)$: custo do caminho desde o nó inicial até o nó n;
- $h(n)$: valor da heurística no nó n.

A partir dessa equação, é possível concluir que o A* se torna equivalente ao UCS quando a parcela $h(n)$ da heurística tem pouco impacto quando somada ao custo do caminho $g(n)$, isto é, quando $g(n)$ é muito maior do que $h(n)$. Nesse caso, $f(n)$ tende a $g(n)$ e o algoritmo se reduz ao UCS.

Nesse sentido, se um mapa tiver custos elevados em todos os nós, a heurística acaba não influenciando tanto a busca, como mostra a Figura 7 (a). Nesse caso, o A* retorna o caminho da Figura 7 (b) expandindo 114 nós, enquanto o UCS retorna o caminho da Figura 7 (c) expandindo 117 nós, praticamente a mesma quantidade.

Figura 6: Caso em que o A* pode ser melhor do que o UCS (heurística euclidiana). (a) Mapa inicial, (b) caminho retornado pelo UCS, e (c) caminho retornado pelo A*.



4 Conclusão

Tendo em vista os objetivos propostos, foi possível implementar os algoritmos de busca em Python e compará-los em diferentes mapas. Por meio dos casos de teste, foi possível verificar as diferenças entre os algoritmos, tanto em termos de qualidade das soluções quanto nos casos em que eles não garantem otimalidade.

Referências

Russell, S.; and Norvig, P. 2020. *Artificial Intelligence: A Modern Approach*. Pearson, 4 edition. ISBN 978-0134610993.

Figura 7: Caso em que o A* é equivalente ao UCS (heurística euclidiana). (a) Mapa inicial, (b) caminho retornado pelo UCS, e (c) caminho retornado pelo A*.