

Relatório Laboratório 02 - Busca Informada

Instituto Tecnológico de Aeronáutica – ITA
Inteligência Artificial para Robótica Móvel – CT-213

Nicholas Scharan Cysne
Turma 22

1. Introdução

Tem-se como objetivo deste laboratório implementar um algoritmo de planejamento de trajetória em grafos, Figura 1. Para encontrar o caminho desejado, utiliza-se os algoritmos Dijkstra, Greedy e A*.

Para a implementação dos três algoritmos foi utilizado uma estrutura auxiliar heap para o mapeamento dos nós já visitados e quais já poderiam ser considerados “fechados”.

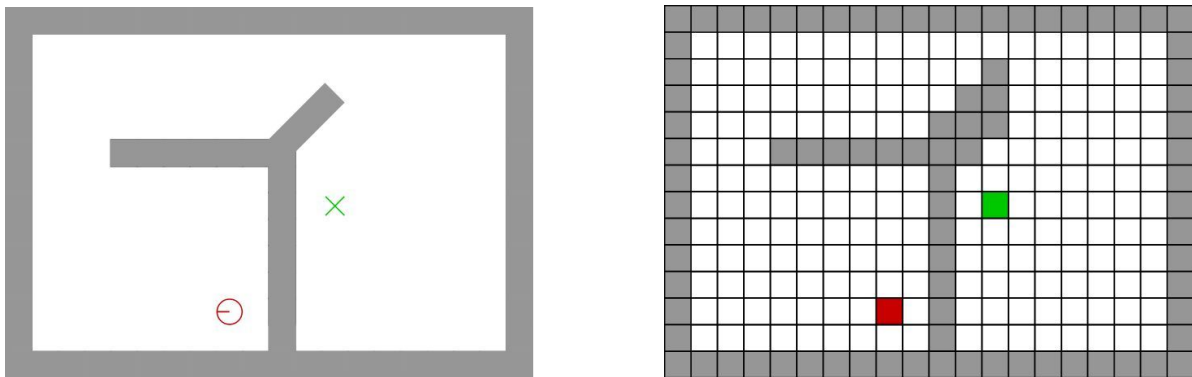


Figura 1: planejamento de caminho de robô móvel usando busca em grafos.

2. Algoritmo Dijkstra

O algoritmo Dijkstra realiza a busca em grafos se baseando em BFS, logo ele encontra os custos necessários para se chegar em todos os vértices de um grafo a partir de uma entrada. O Algoritmo pode ser otimizado para parar sua execução assim que encontra o caminho mínimo ao ponto de destino, deixando de ser um algoritmo de mapeamento para ser um de busca.

Este algoritmo sempre encontra a solução ótima pois considera a distância mínima entre dois pontos através dos pontos que já visitou. A Figura 2 mostra o caminho encontrado pelo Dijkstra para o primeiro problema gerado pelo código.

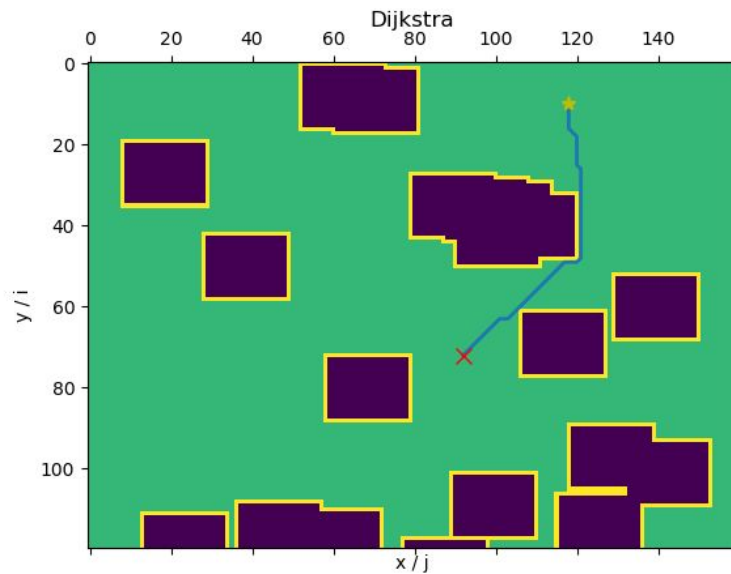


Figura 2: Resultado Dijkstra para a primeira iteração do gerador de caminhos.

3. Algoritmo Greedy

O algoritmo Greedy, similar ao Dijkstra, realiza a busca em grafos se baseando em BFS, entretanto este opera apenas através de heurísticas, realizando “chutes” acerca da distância real dos nós atuais ao destino e seguindo aqueles que possuem a menor heurística.

Este algoritmo encontra normalmente a solução subótima pois considera somente as heurísticas entre o ponto atual e o objetivo. A Figura 3 mostra o caminho encontrado pelo Greedy para o primeiro problema gerado pelo código.

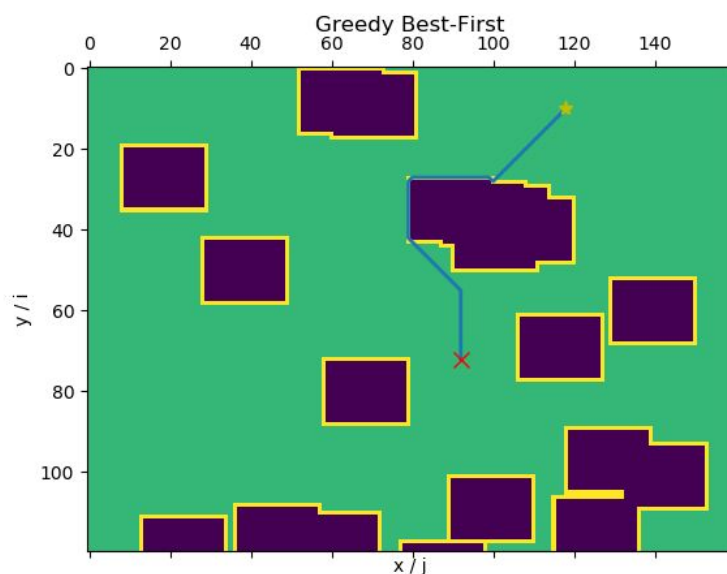


Figura 3: Resultado Greedy para a primeira iteração do gerador de caminhos.

4. Algoritmo A*

O algoritmo A*, similar aos outros dois, realiza a busca em grafos se baseando em BFS e pode ser considerado uma junção dos dois algoritmos anteriores, onde há a contagem do caminho já realizado e dos nós já visitados com a heurística do caminho que falta. Assim o A* possui uma certa noção do caminho a ser traçado.

Este algoritmo encontra a solução ótima pois considera o caminho já realizado e as heurísticas entre o ponto atual e o objetivo. A Figura 4 mostra o caminho encontrado pelo A* para o primeiro problema gerado pelo código.

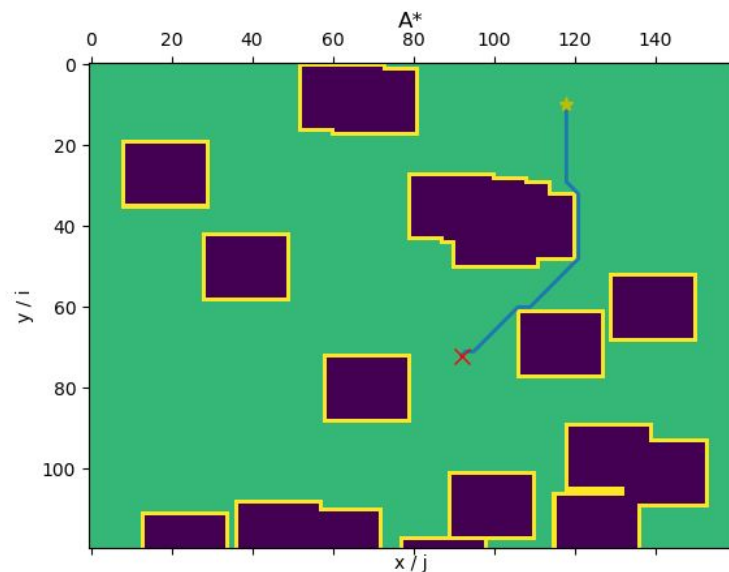


Figura 4: Resultado A* para a primeira iteração do gerador de caminhos.

5. Dados Comparativos

A Tabela 1 representa os valores médios e desvio padrão obtidos após a execução do Monte Carlo, ou seja 100 iterações do algoritmo, pelo computador. Foi utilizado um notebook com as seguintes configurações:

- Processador Intel i7 8a Geração
- 12Gb Memória RAM
- 2Gb Memória dedicada a vídeo

Algoritmo	Tempo computacional (s)		Custo do Caminho	
	Média	Desvio Padrão	Média	Desvio Padrão
Dijkstra	0.50680	0.23718	80.58818	34.89442
Greedy Search	0.07090	0.03256	107.59062	59.38319
A*	0.15559	0.10542	80.58818	34.89442

Tabela 1: tabela de comparação entre os algoritmos de planejamento de caminho.

Nota-se que A* e Dijkstra encontraram caminhos com custos iguais, os dois encontraram soluções ótimas iguais, porém o tempo computacional do A* é cerca de três vezes inferior. O algoritmo Greedy apesar de possuir o tempo computacional mais baixo encontra soluções subótimas, diminuindo sua eficiência.

As Figuras 5, 6 e 6 mostram outras iterações dos três algoritmos, podendo notar a diferença na execução de cada um.

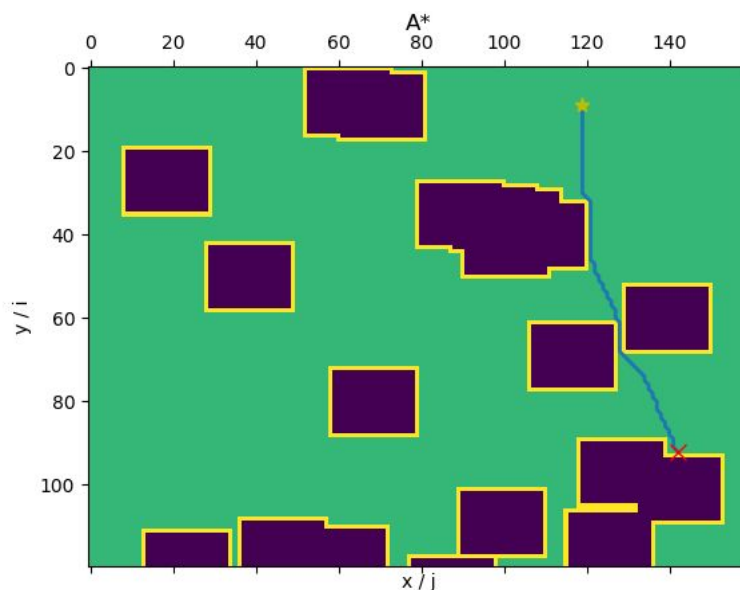


Figura 5: 4ª Iteração do Algoritmo A*.

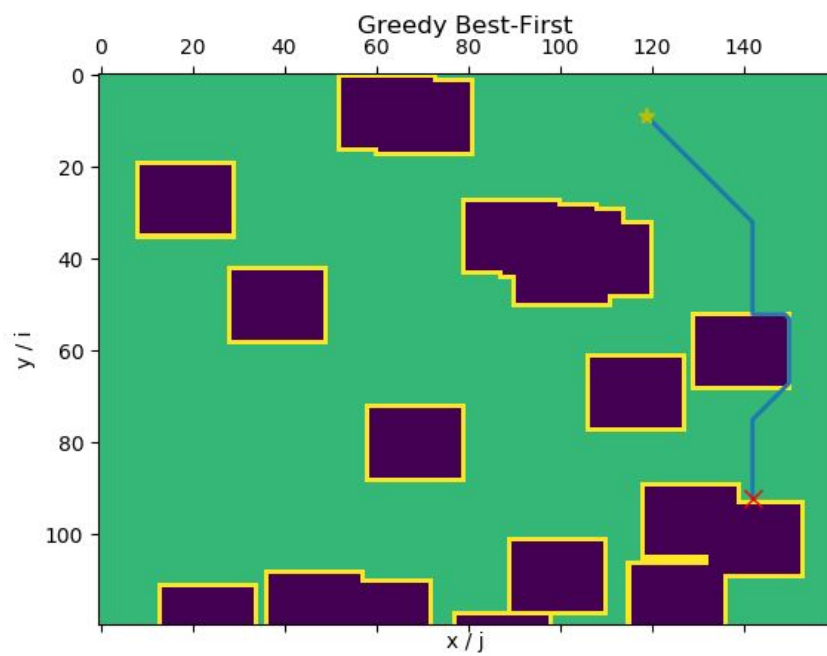


Figura 6: 4ª Iteração do Algoritmo Greedy.

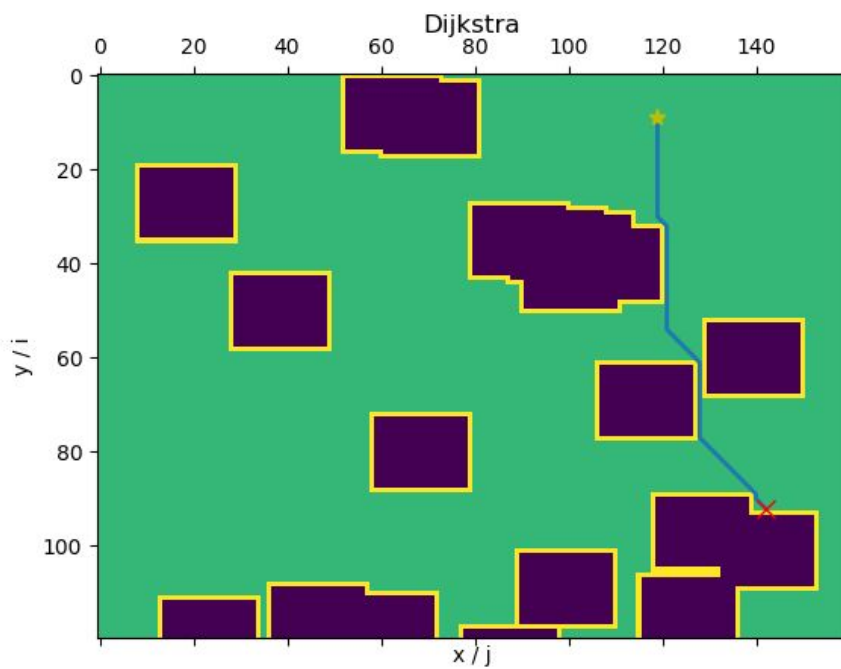


Figura 7: 4ª Iteração do Algoritmo Dijkstra.