

Busca Informada

Samara Ribeiro Silva

Instituto Tecnológico de Aeronáutica, Laboratório de Inteligência Artificial para Robótica Móvel (CT-213). Professor Marcos Ricardo Omena de Albuquerque Máximo, São José dos Campos, São Paulo, 22 de março de 2021.

Os algoritmos de busca utilizados foram o Dijkstra, A* e Greedy. Os algoritmos foram implementados conforme ensinado em aula, adicionando apenas a variável *closed* para indicar que o *node* já havia sido finalizado para suprir a falta de uma função para editar o *node* na fila de prioridade.

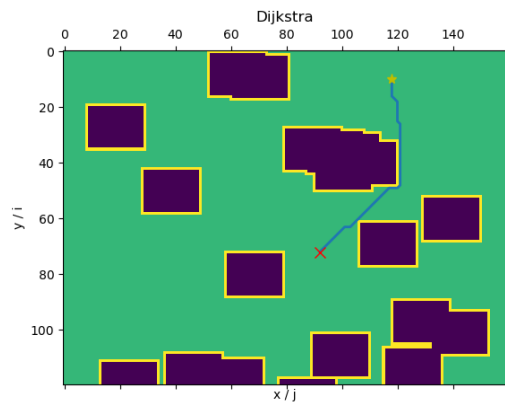
O algoritmo Dijkstra verifica o custo de todas as arestas do nó e caso o custo acumulado (custo para percorrer do início até o nó em questão) do nó sucessor C_s seja maior que o custo do nó analisado C somado com o custo da aresta C_a o custo do nó sucessor é atualizado para $C_s = C + C_a$. Para esse laboratório não foi necessário encontrar todos os custos acumulados, portanto o programa retorna o caminho e o custo do caminho quando chega no nó objetivo. Vale ressaltar que esse algoritmo encontra a solução ótima.

Já o algoritmo A* utiliza uma estimativa de custo total do caminho utilizando uma função heurística $h(n)$. Essa função $h(n)$ é dependente do problema e deve ser admissível, ou seja, a estimativa ser otimista para garantir a solução ótima. A grande diferença entre o A* e o Dijkstra é o Custo do nó sucessor. Para realizar a atualização do valor do nó sucessor no A* utilizamos que o custo do nó sucessor deve ser maior que o custo acumulado do nó analisado somado com o custo da aresta e o valor da função heurística para o nó sucessor: se $C_s > C + C_a + h(n_s)$ o valor de C_s é atualizado para $C_s = C + C_a + h(n_s)$.

O algoritmo de busca Greedy encontra uma solução subótima e difere dos outros dois pois considera primeiro os nós mais promissores através de uma função heurística ou seja a valor da prioridade utilizada como critério para a saída da fila de prioridade é o valor da função heurística.

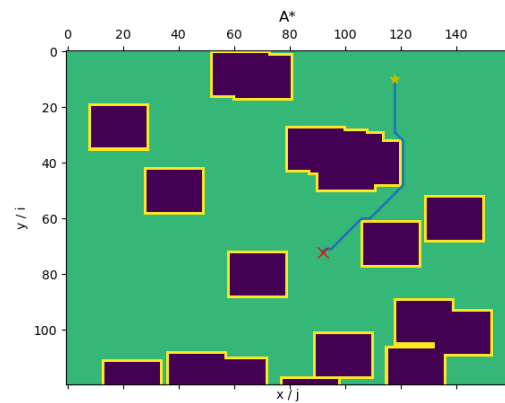
Nas figuras 1, 2 e 3 é possível observar os resultados dos algoritmos implementados com os valores do custo computacional e o custo do caminho encontrado. Na tabela 1, são mostrados os resultados dos algoritmos com $n = 100$.

Figura 1:Dijkstra



Compute time: 0.1693
Cost: 78.1837

Figura 2: A*



Compute time: 0.0533
Cost: 78.1838

Figura 3: Greedy



Compute time: 0.0156
Cost: 135.3259

Tabela 01: Resultados obtidos após a simulação de Monte Carlos com número de repetições $n = 100$.

Algoritmo	Tempo computacional (s)		Custo do caminho	
	Média	Desvio padrão	Média	Desvio padrão
Dijkstra	0.245306	0.245306	79.829197	38.570924
Greedy Search	0.008869	0.007524	103.34198	59.409721
A*	0.064373	0.063546	79.829197	38.570962