Sistemas Robóticos Autônomos

Algoritmo A*

Busca do Menor Caminho

- Muitos métodos de planejamento se baseiam na ideia de capturar a conectividade do espaço livre na forma de um grafo de conectividade. Exemplo: Mapas de Rotas e Decomposição em Células Convexas.

- A busca de um caminho entre q_{ini} e q_{fin} é simplificada, ficando reduzida à busca de um caminho no grafo entre os nós correspondentes N_{ini} e N_{fin}.

- Técnicas padrão de busca em grafos podem ser utilizadas. Exemplo: Algoritmo A^*.

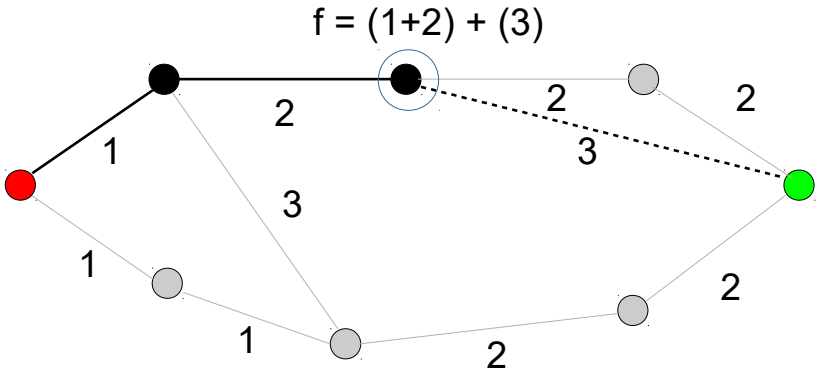
Algoritmo A*

- Aplicável a grafos em que os arcos têm custos associados, por exemplo, distância entre os nós.
- Custo de um caminho = soma dos custos dos seus arcos.
- Permite obter o caminho de menor custo.
- Complexidade: $O(r.\log(n))$, r = Nº de arcos, n = Nº de nós.

Função de Custo

Função de Custo: $f(N) = g(N) + h(N)$

- $g(N)$ = custo do caminho entre N_{ini} e N em T corrente.
- $h(N)$ = estimativa heurística do custo $h^*(N)$ do caminho de mínimo custo entre N e N_{fin}. Exemplo: $h(N)$ = distância Euclideana, $h(N) = 0$ (Dijkstra).



Algoritmo A*- Procedimento

- G explorado iterativamente a partir de N_{ini}.
- Caminhos gerados formam árvore T do subconjunto de G já explorado.
- T representada através de ponteiros, de cada nó visitado ao seu nó pai.
- Para cada nó N, atribui-se uma função de custo f(N).
- Escolhe-se o nó corrente como o de menor custo entre os já explorados.
- A partir do no corrente exploram-se os seus nós vizinhos.
- Para cada nó visitado, um ou mais caminhos são gerados a partir de N_{ini}, mas só o de menor custo é memorizado.

Algoritmo A*- Procedimento

Estruturas de Dados:

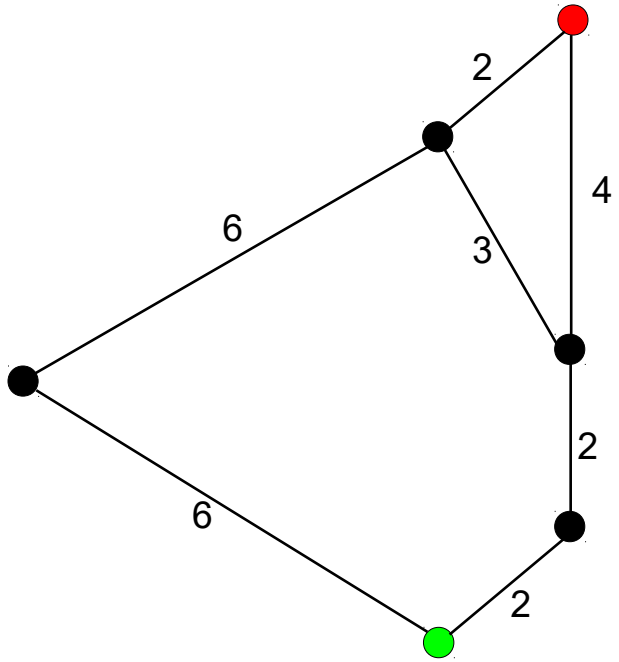
- G(X,A) = Grafo com n nós $\in$ X e r arcos $\in$ A.
- Conectividade de G representada por listas de nós vizinhos a cada nó.
- T = Árvore do subconjunto de G já visitado.
- L = Lista que armazena nós de G ordenados por f(N).
- $k : X \times X \rightarrow \mathbb{R}^+$ função que especifica o custo de cada arco.
- h(N): estimativa do custo do caminho mínimo entre N e N_{fin}.
- g(N) = custo do caminho entre N_{ini} e N em T corrente.

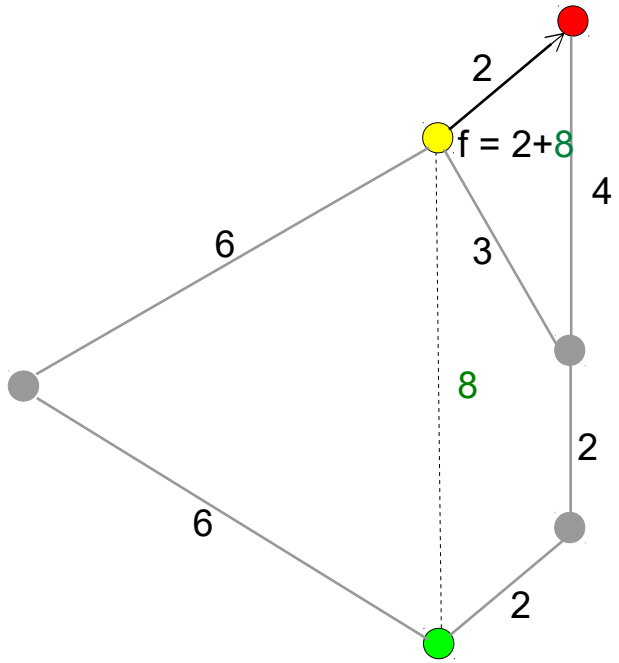
Algoritmo A*- Procedimento

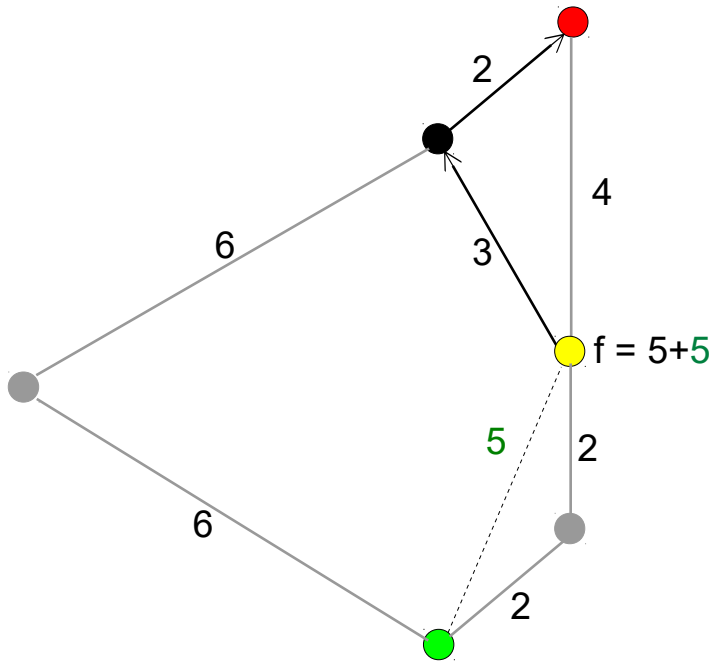
Operações sobre a lista L:

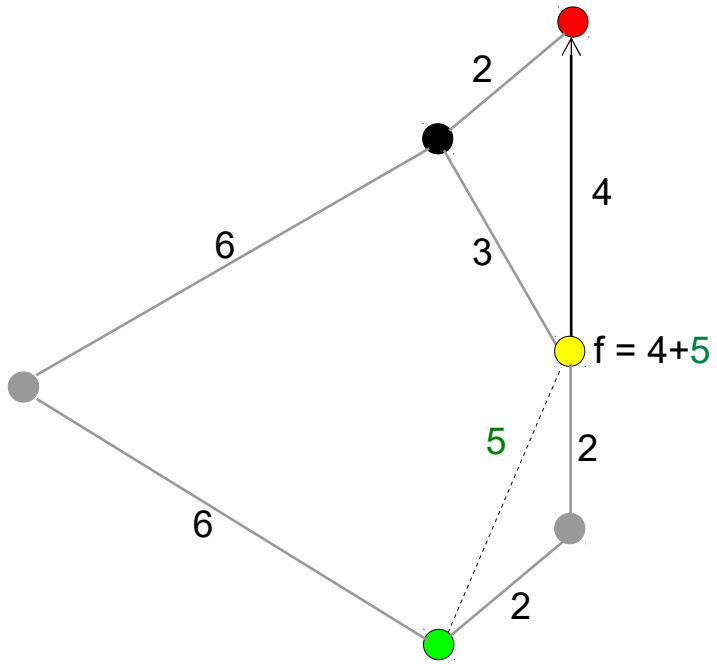
- PRIMEIRO(L): retorna o nó com menor valor de f(N) em L e o remove da lista.
- INSERIR(N,L): insere o nó N na lista L.
- APAGAR(N,L): apaga o nó N da lista L.
- MEMBRO(N,A): determina se o nó N é membro da lista L.
- VAZIA(L): determina se a lista L está vazia.

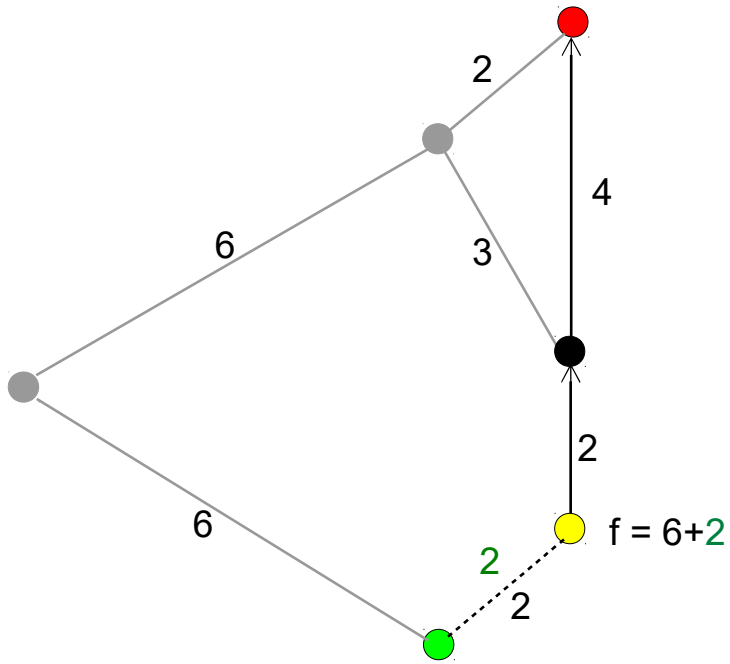
Procedimento $A^*(G, N_{ini}, N_{fin}, k, h)$;
começar
N_{ini} em T;
INSERIR(N_{ini}, L); marcar N_{ini} como visitado;
enquanto ¬VAZIA(L), **faça**
começar
N ← PRIMEIRO(L);
se N = N_{fin}, **então** sair do laço while;
para cada nó N' adjacente a N em G, **faça**
se N' é não visitado, **então**
começar
adicionar N' a T com um ponteiro para N;
INSERIR(N',L); marcar N' como visitado;
fim
se não, se g(N') > g(N) + k(N, N'), **então**
começar
redirecionar o ponteiro de N' para N em T;
se MEMBRO(N',L),**então** APAGAR(N',L);
INSERIR(N',L);
fim
fim;
se ¬VAZIA(L), **então**
retornar o caminho traçando os ponteiros de N_{fin} a N_{ini};
se não reportar falha;
fim;

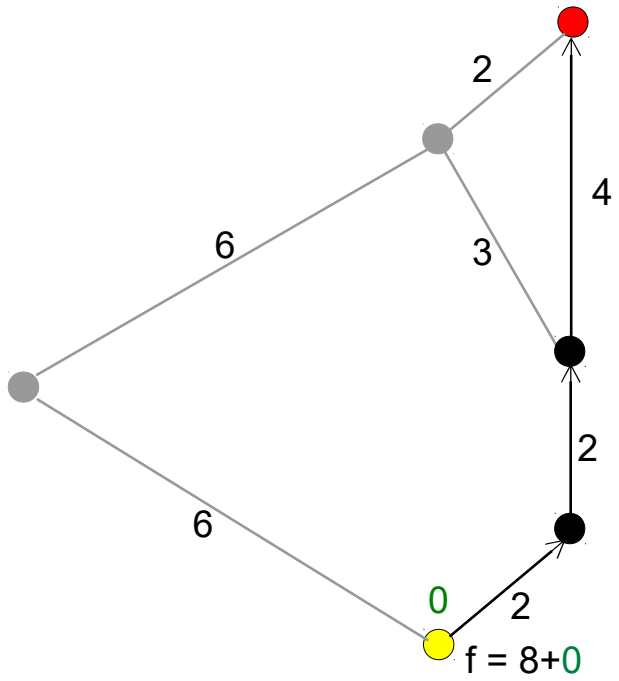

2
4
6
3
2
6
2

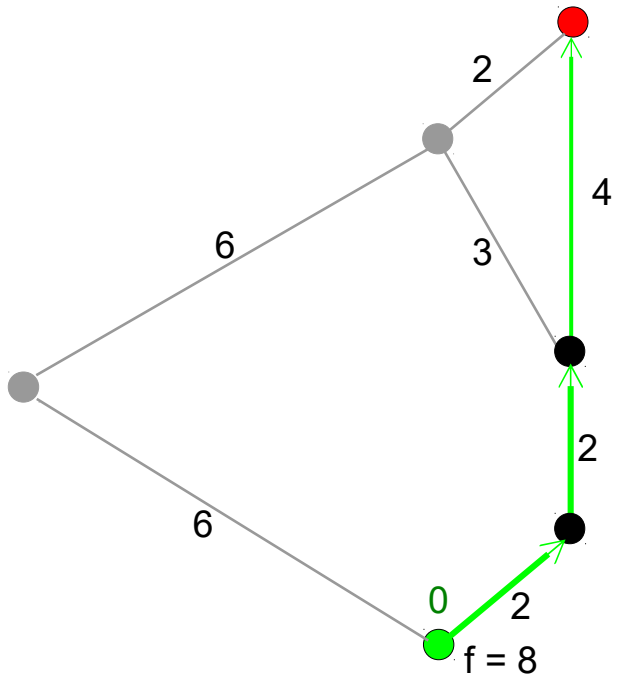

2
f = 2+8
4
6
3
8
2
6
2


2
4
6
3
f = 5+5
5
2
6
2


2
4
6
3
f = 4+5
5
2
6
2


2
4
6
3
2
6
f = 6+2
2
2


2
4
6
3
2
6
0
2
f = 8+0


2
4
6
3
2
6
0
2
f = 8

Sistemas Robóticos Autônomos

Algoritmo A*