

Algoritmo A* - Prof. João Alberto Fabro

Utiliza as seguintes funções:

g = função de avaliação do custo par chegar até o nó (soma dos custos da aplicação de cada regra que leva do estado inicial até este nó.)

h' = função de avaliação que estima o custo para se chegar deste nó até o estado final. Esta função é a responsável pela parte “heurística” do processamento. É chamada de h' para indicar que é uma aproximação da função de avaliação exata h do custo para chegar deste nó até o estado final.

$f' = g + h'$ – É a avaliação global de um determinado nó.

E as seguintes listas:

ABERTOS: nós que foram gerados e sofreram a aplicação da função heurística, mas que ainda não foram examinados(isto é, seus sucessores ainda não foram gerados). ABERTOS é na verdade uma fila de prioridades onde os elementos com prioridade máxima são aqueles onde a avaliação tem o valor mais promissor. Técnicas padrões que manipulam filas de prioridades podem ser utilizadas para manipular esta lista.

FECHADOS: nós que já foram examinados. Precisamos manter também estes nós na memória se quisermos efetuar uma busca em grafo em vez de em árvore, já que sempre que um nó é gerado, precisamos verificar se ele já foi gerado antes (se está em ABERTOS, ou se já esteve em ABERTOS, e agora está em FECHADOS).

Algoritmo:

- 1) Comece com ABERTOS contendo apenas o nó inicial. Atribua 0 ao valor de g do nó, e seu valor de h' para qualquer que seja(dado pela função heurística), e atribua $f' = g + h'$. Atribua a FECHADOS a lista vazia.
- 2) Até ser encontrado um nó-meta, repita o seguinte procedimento: se não houver nenhum nó em ABERTOS, reporte fracasso e saia. Caso contrário, pegue o nó de ABERTOS com menor valor de f' . Remova-o de ABERTOS. Coloque-o em FECHADOS. Veja se o MELHOR-NÓ é um nó –meta. Em caso positivo, encerre e reporte a solução(MELHOR-NÓ se tudo o que quisermos for o nó, ou o caminho que foi criado entre o estado inicial e MELHOR-NÓ se estivermos interessados no caminho). Caso contrário, gere os sucessores de MELHOR-NÓ mas ainda não ajuste MELHOR-NÓ para apontar para eles. (Precisamos primeiro ver se algum deles já foi gerado.) Para cada SUCESSOR, faça o seguinte:
 - (a) Ajuste SUCESSOR para apontar de volta para MELHOR-NÓ. Estas ligações para trás permitirão recuperar o caminho, uma vez encontrada a solução.
 - (b) Compute $g(\text{SUCESSOR}) = g(\text{MELHOR-NÓ}) + \text{custo de sair de MELHOR-NÓ e chegar a SUCESSOR}$.
 - (c) Veja se SUCESSOR é igual a qualquer nó em ABERTOS(isto é, se ele já foi gerado, mas ainda não processado). Em caso positivo, chame aquele nó de ANTIGO. Uma vez que este nó já existe no grafo, podemos desconsiderar SUCESSOR e acrescentar ANTIGO à lista de sucessores de MELHOR-NÓ. Agora precisamos decidir se a ligação com o pai de ANTIGO deve ser ou não reajustada para apontar para MELHOR-NÓ. Ela deve ser reajustada se o caminho que encontramos para chegar a SUCESSOR for mais barato que o atual melhor caminho até ANTIGO(já que SUCESSOR e ANTIGO são, na realidade, o mesmo nó). Portanto, veja se é mais barato chegar a ANTIGO via seu pai atual ou a SUCESSOR via MELHOR-NÓ: compare os valores de g . Se ANTIGO for mais barato(ou se o preço for o mesmo), então não precisaremos fazer nada. Se SUCESSOR for mais barato, então reajuste a ligação com o pai de ANTIGO, para apontar para MELHOR-NÓ, registre o novo caminho mais barato em $g(\text{ANTIGO})$ e atualize $f'(\text{ANTIGO})$.
 - (d) Se SUCESSOR não estava em ABERTOS, veja se está em FECHADOS. Em caso positivo, chame o nó em FECHADOS de ANTIGO e acrescente ANTIGO à lista de sucessores de MELHOR-NÓ. Verifique se o novo caminho ou o caminho antigo é melhor, como na etapa 2(c) e ajuste corretamente a ligação com o pai e os valores de g e f' . Se tivermos acabado de encontrar um melhor caminho para ANTIGO, precisamos propagar a melhoria aos sucessores de ANTIGO. Essa operação é um pouco delicada. ANTIGO aponta para seus sucessores. Cada sucessor, por sua vez, aponta para seus sucessores, e assim por diante, até que cada ramificação acabe com um nó que ainda esteja em ABERTOS ou que não tenha sucessores. Portanto, para propagar um novo custo a todos os sucessores, faça caminhamento em profundidade da árvore, começando em ANTIGO, mudando o valor de g em cada nó (e portanto também o valor de f'), e encerrando cada ramificação quando encontrar um nó sem sucessores ou um nó para o qual um caminho melhor ou equivalente já tenha sido encontrado(*). Esta condição é fácil de ser verificada. A ligação de cada nó com o pai aponta para o melhor pai conhecido. No momento da propagação, verifique se o pai aponta para o nó do qual você está vindo. Se sim, continue. Caso contrário, então o valor de g já reflete o melhor caminho do qual ele é parte. Então a propagação pode parar por aqui. Mas é possível que, com o novo valor de g sendo propagado aos sucessores, o percurso em que você esteja torne-se melhor do que o caminho através do pai atual. Então compare os dois. Se o caminho através do pai atual ainda for melhor, encerre a propagação. Se o caminho através do qual você está propagando agora for melhor, reajuste o pai e prossiga com a propagação.
 - (e) Se SUCESSOR não estava nem em ABERTOS nem em FECHADOS, então coloque-o em ABERTOS e acrescente-o à lista de sucessores de MELHOR-NÓ. Compute $f'(\text{SUCESSOR}) = g(\text{SUCESSOR}) + h'(\text{SUCESSOR})$.
- (*) Essa Segunda verificação garante que o algoritmo termine mesmo que haja ciclos no grafo. Se houver um ciclo, quando um dado nó for visitado pela Segunda vez, o caminho não será melhor que na primeira vez, portanto, a propagação acabará.