

Especificação

Você deverá resolver um problema simplificado de entregas e empacotamento em um depósito, modelado a partir do problema bem mais complexo que os robôs da Amazon devem enfrentar todos os dias. Veja como funciona o problema real em <https://www.youtube.com/watch?v=6KRjuuEVEZs>. O depósito tem vários produtos, um conjunto de estações de empacotamento e um conjunto de robôs. Os produtos e as estações de empacotamento estão em localizações fixas espalhadas pelo depósito. Os robôs podem se mover pelo depósito. Um problema típico neste domínio é operar os robôs para que eles completem um conjunto de pedidos. Cada pedido requer que um produto em particular seja movido para uma estação de empacotamento. Um robô deve ser associado a esse trabalho e ele se moverá para a localização do produto, pegará o produto e então se moverá para a estação de empacotamento para entregar o produto. Depois da entrega, o robô permanecerá na estação de empacotamento, até que ele seja associado a um outro trabalho. O problema é resolvido quando todos os pedidos forem associados aos robôs e todos os robôs tenham completado todos os trabalhos para os quais eles foram associados. A qualidade da solução é medida de acordo com o tempo que a entrega final é realizada. Ou seja, uma solução melhor entrega os pedidos para as estações de empacotamento mais rápido. O problema é simplificado em diversas maneiras, comparado com a aplicação real. Algumas dessas simplificações incluem: o robô só pode carregar um produto de cada vez; cada pedido consiste de apenas um produto; não nos preocupamos com as ações envolvidas em pegar o produto e colocá-lo na estação de empacotamento—só pedimos para o robô pegar o produto em uma posição (x_1, y_1) e entregá-lo na posição (x_2, y_2) , sem se preocupar com os passos intermediários.

Posições e movimentos do robô

Posições são especificadas usando um sistema de coordenadas simples x - y , em que x e y são inteiros, maiores ou iguais a zero. Os robôs podem se mover entre quaisquer duas posições. Se ele está correntemente na posição (x_1, y_1) ele pode ir para a posição (x_2, y_2) . Ele só se move na horizontal ou vertical e gasta uma unidade de tempo para se mover em qualquer direção, uma posição para a frente ou para trás, para a direita ou para a esquerda. Por exemplo, para se mover da posição $(1,2)$ para $(4,5)$, ele precisará de $\text{abs}(1-4) + \text{abs}(2-5) = 6$ unidades de tempo. Ou seja, ele precisa se mover $+3$ na direção x e $+3$ na direção y .

Concorrência

Quando temos vários robôs, temos que gerenciá-los executando entregas concorrentemente. Para conseguir fazer isso, os estados do espaço de estados incluirão (a) tempo corrente e (b) alguns eventos pendentes que acontecerão no futuro (mas apenas o futuro desse estado em particular - cada estado tem seus próprios eventos futuros). Isso permitirá que tenhamos um espaço de estados em que múltiplas atividades concorrentes estão acontecendo em qualquer estado. Todas essas atividades tem um fim em algum

tempo no futuro. Os eventos futuros serão todos entregas completas de diferentes robôs, tal que temos um estado onde um grupo de robôs estão trabalhando concorrentemente em diferentes entregas. Cada entrega estará completa em algum tempo (diferente) no futuro. Para gerenciar tais estados, temos dois tipos diferentes de ações: (1) ações que iniciam atividades e atualizam o estado para incluir a informação sobre o tempo futuro quando estas atividades estarão completas. Esse tipo de ação gera um novo estado que terá o mesmo tempo que o estado corrente. Ou seja, esse tipo de ação não move o tempo para a frente e custa zero (uma vez que estamos tentando encontrar uma solução para completá-la no tempo mais cedo possível); (2) uma ação simples que move o tempo para a frente. Esta ação opera em um estado movendo o tempo para a frente, para o instante de tempo em que o evento futuro mais próximo (temporalmente) estará terminado. Estes eventos envolvem robôs terminando suas entregas. Essas ações custam uma quantidade igual a quantidade de tempo que foi movido.