

Solução de Problemas por Busca

Espera-se que os agentes inteligentes façam com que o ambiente passe por uma seqüência de estados (através das suas ações) que maximize a medida de desempenho. Esta tarefa é bastante facilitada se o agente puder usar, além de suas percepções, um objetivo (goal) e busque atingi-lo.

Agentes resolvedores de problemas tentam encontrar seqüências de ações que levam ao estado desejado.

A primeira etapa na solução de problemas é a formulação do objetivo, baseada no estado atual do ambiente. O objetivo direciona o comportamento do agente.

A segunda etapa é a formulação do problema, que é o processo de decidir que ações e estados considerar.

Em geral, um agente com várias opções imediatas, de valor desconhecido, pode decidir o que fazer primeiro examinando várias seqüências possíveis de ações que levem a estados de valor conhecido, e escolher a melhor.

Exemplo: imagine um agente na cidade romena de Arad, e que queira ir para Bucareste. Há três estradas que partem de Arad, que vão para Sibiu, Timisoara e Zerind. Nenhuma delas vai até Bucareste. Que estrada ele deve escolher?

O processo de procurar uma seqüência de ações que levem ao objetivo é chamado de Busca, que é a terceira etapa.

Um algoritmo de busca recebe um problema como entrada e retorna uma solução em forma de seqüência de ações.

A quarta e última fase é a execução, onde as ações recomendadas são executadas na ordem indicada.

1. Formulação do objetivo
2. Formulação do problema
3. Procedimento de busca
4. Execução da solução

Formulação do problema

Há quatro tipos essenciais de problemas:

- Problema de estado único
- Problema de múltiplos estados
- Problema de possibilidade
- Problema de exploração

Mundo do aspirador: Há só dois lugares no mundo. Cada lugar pode ou não conter pó e o agente pode estar em um lugar ou no outro. O agente tem três ações possíveis: Esquerda, Direita e Aspirar. A aspiração é 100% efetiva e o objetivo é limpar todo o pó.

Problema de estado único

Ocorre quando os sensores do agente lhe dão informações suficientes para que ele saiba exatamente em que estado ele está (ou seja, o mundo é acessível). Além disso, o agente sabe exatamente o que cada uma de suas ações faz.

Assim, ele pode calcular exatamente em que estado ele vai estar após a seqüência de quaisquer ações.

Por exemplo, se seu estado inicial for 5, ele calcula que a seqüência [Direita, Aspirar] atinge o objetivo.

Problema de múltiplos estados

Esse tipo de problema aparece quando o agente sabe os efeitos de suas ações, mas tem acesso limitado ao estado do mundo. Por exemplo, no caso mais extremo, ele pode não ter nenhum sensor. Neste caso, ele somente sabe que seu estado inicial está no conjunto {1, 2, 3, 4, 5, 6, 7, 8}.

Como o agente sabe o que suas ações fazem, ele pode, por exemplo, calcular que a ação Direita vai colocá-lo em um dos estados {2, 4, 6, 8}. Assim, ele pode ver que a seqüência de ações [Direita, Aspirar, Esquerda, Aspirar] chega com certeza ao objetivo, não importando o estado inicial.

Em resumo: quando o mundo não é totalmente acessível, o agente deve raciocinar em termos de conjuntos de estados que ele pode chegar, ao invés de um único estado.

O caso da ignorância a respeito do resultado das ações pode ser tratado de forma semelhante (Lei de Murphy). Por exemplo, suponhamos que a ação Aspirar não seja 100% e que as vezes coloque pó no chão, se ele já estiver limpo. Se o agente estiver no estado 4, ele sabe que, se usar a ação Aspirar, vai acabar em um dos estados {2,4}.

Suponha um agente no mundo do aspirador de Murphy, que tem um sensor de sujeira local e um sensor localizador.

Por exemplo, os sensores podem dizer que ele está em um dos estados {1,3}. O agente pode então formular uma seqüência de ações {Aspirar, Direita, Aspirar}. A primeira ação, Aspirar, o coloca em um estado entre {5, 7}; mover em seguida para a Direita o coloca em um estado entre {6, 8}. Se for o estado 6, a seqüência vai funcionar; senão, pode falhar, pois deveria ter parado em 8. Se ele escolher, de início, uma seqüência [Aspirar], nem sempre vai ter sucesso.

Problema de possibilidade

Há uma maneira de resolver o problema partindo de {1,3}, que seria: primeiro Aspirar, depois mover para a Direita e Aspirar somente se houver sujeira naquele local.

Portanto, resolver o problema requer ler os sensores durante a fase de execução. O agente tem que calcular não mais uma seqüência de ações, mas sim uma árvore interia de ações. Em geral, cada ramo da árvore lida com uma possibilidade que possa surgir durante a execução.

Muitos problemas no mundo real são de possibilidade; e é por isso que a maioria das pessoas anda de olhos abertos.

Problema de exploração

Surge quando o agente não tem nenhuma informação sobre os efeitos de suas ações. Isto é como estar perdido em um país estranho (com idioma estranho), sem mapa e é a tarefa mais difícil enfrentada por um agente inteligente.

Neste caso, o agente deve experimentar, gradualmente descobrindo o que suas ações fazem e que estados existem. Fazer uma ação pode resolver um perigo significativo para um agente ignorante. Se sobreviver, o agente aprende um “mapa” do ambiente, que pode usar mais tarde para resolver problemas subseqüentes.

Problemas bem definidos

Um problema é um conjunto de informações que o agente vai usar para decidir o que fazer. Em um problema de estado único, essas informações são:

- O estado inicial, onde o agente sabe que está;
- Um conjunto de ações possíveis disponíveis para o agente. O termo operador é usado para denotar a descrição de uma ação e termos de saber que estado será atingido por tomar a ação em cada estado;
- Um teste de objetivo, que o pode ser aplicado à descrição de um estado para determinar se este é um estado final.

Os dois primeiros itens definem o espaço de estados do problema: o conjunto de todos os estados alcançáveis, a partir do estado inicial, por qualquer seqüência de ações.

Um caminho dentro do espaço de estados é simplesmente qualquer seqüência de ações levando de um estado a outro.

Uma solução pode ser preferível em relação a outra, mesmo que ambas atinjam o objetivo. Ex: menos ações.

O custo de um caminho (g) é uma função que atribui um valor a um caminho, geralmente a soma dos custos das ações individuais tomadas ao longo do caminho.

Medidas de Desempenho

A eficiência de uma busca pode ser medida em três maneiras:

- I) se ela encontrou uma solução;
- II) se sim, foi uma boa solução (como baixo custo de caminho);
- III) qual foi o custo de busca associado ao tempo de CPU e memória necessários para encontrar uma solução.

O custo total de uma busca (uma solução específica) é a soma do custo do caminho e do custo de busca.

No problema da viagem na Romênia, o custo do caminho pode ser proporcional aos Km. O custo de busca depende do ambiente: se for estático, é zero; senão, pode ser ms.

Exemplos de problemas

A quantidade de ambientes que podem ser caracterizados como problemas bem definidos é enorme. Os problemas de brinquedo (toy problems) são usados para ilustrar técnicas de solução de problemas e os problemas do mundo real (real world problems) tendem a ser mais difíceis e suas soluções podem ser usadas na prática.

Exemplos de toy problems: puzzle, as oito rainhas, mundo do aspirador, etc.

Exemplo de real world problems: busca de rotas, problema do caxeiro viajante, navegação de robôs, etc.

Referências:

RUSSELL, Stuart J. Inteligência artificial, Ed. Elsevier, 2004, 2ª edição