

Universidade Federal de São João del-Rei
Inteligência Artificial
Missionários e Canibais

Felipe Francisco Rios de Melo
Heitor Lourenço Werneck
Thales Mrad Leijoto

1. Introdução

No âmbito da Inteligência Artificial, um problema consiste em cinco partes: o estado inicial, um conjunto de ações, um modelo de transição descrevendo os resultados dessas ações, uma função teste de objetivo e uma função custo de caminho. O ambiente do problema é representado por um espaço de estados. Um caminho pelo espaço de estados a partir do estado inicial até um estado objetivo é uma solução.

Algoritmos de busca tratam estados e ações como atômicos: sem considerar qualquer estrutura interna que possam ter.

Métodos de busca não informados têm acesso apenas à definição do problema. Dentre estes métodos, temos a busca por aprofundamento iterativo. A busca de aprofundamento iterativo chama a busca em profundidade com limites crescentes de profundidade até encontrar um objetivo. Ela é completa, ótima para passos de custo unitário, tem complexidade de tempo comparável à busca em largura e tem complexidade de espaço linear.

Este trabalho, consiste no estudo do problema dos missionários e canibais, bem como uma implementação, utilizando o algoritmo de busca por aprofundamento iterativo para encontrar uma solução para o problema.

2. Formulação e modelagem do problema

O problema de missionários e canibais é normalmente enunciado como a seguir:

Três missionários e três canibais estão em um lado de um rio, juntamente com um barco que pode levar uma ou duas pessoas. Descubra um meio de fazer todos atravessarem o rio sem deixar que um grupo de missionários de um lado fique em número menor que o número de canibais nesse mesmo lado do rio.

2.1. Formulação do problema

a.1) *Formule o problema precisamente, fazendo apenas as especificações necessárias para assegurar uma solução válida.*

2.1.1. Estado inicial

O estado é denotado pela quantidade de missionários, canibais e barco em cada margem.

Para representar o estado do mundo será utilizado uma tupla de tamanho 3 para uma margem do rio. A tupla terá a informação da quantidade de missionários, canibais e barco (se o barco está na margem). Ou seja:

Margem inicial = (*missionários, canibais, barco*)

É importante notar que somente com uma margem representada é possível saber o estado da outra margem, pois o número de missionários (canibais) deve ser não negativo e menor ou igual a 3 e somente um barco existe.

O estado inicial possui 3 missionários, 3 canibais e 1 barco em uma margem.

Margem inicial = (3, 3, 1)

2.1.2. Ações

Em cada travessia de barco pode-se mover: 1 missionário; 1 canibal; 1 canibal e 1 missionário; 2 canibais ou 2 missionários para a outra margem.

Deve-se assegurar que em ambas as margens o número de missionários deve ser maior ou igual ao número de canibais.

O barco também só se move com 1 ou mais agentes nele.

O número de missionários, canibais e barco não pode ser menor que 0.

O barco somente embarca agentes da sua respectiva margem.

2.1.3. Modelo de transição

Dado o estado e a ação, realizar a travessia de uma margem para outra, devolvendo no fim o estado resultante.

2.1.4. Teste de objetivo

A solução deve ter a configuração (0, 0, 0), ou seja, todos missionários, canibais e o barco devem estar na outra margem.

2.1.5. Custo de caminho

Devido ao fato de o problema dos missionários e canibais não envolver custos distintos para ações distintas, foi definido custo 1 para qualquer ação realizada. O custo total de caminho é dado pela quantidade de travessias realizadas até alcançar a solução do problema. Então o problema será de encontrar o menor caminho do estado inicial até o objetivo.

2.2. Descrição do ambiente de tarefas

Ao projetar um agente, a primeira etapa deve ser sempre especificar o ambiente de tarefa de forma tão completa quanto possível.

Abaixo, temos o *PEAS* (Performance, Environment, Actuators, Sensors — desempenho, ambiente, atuadores, sensores), e as características do ambiente de tarefas.

TABELA 1 - Descrição de *PEAS* do ambiente de tarefas para o problema dos Os missionários e os canibais.

Tipo de agentes	Medida de desempenho	Ambiente	Atuadores	Sensores
3 missionários e 3 canibais	Manter todos os missionários vivos, realizar o mínimo possível de travessias, terminar com todos os agentes na margem oposta à margem do estado inicial	Rio, barco e margens	Mover o barco, entrar/sair do barco e atacar (atuador específico dos canibais)	Número de missionários e canibais em cada margem, número de indivíduos no barco

TABELA 2 - Ambiente de tarefas e suas características.

Observável	Agentes	Determinístico / Estocástico	Episódico / Sequencial	Estático / Dinâmico	Discreto / Contínuo	Conhecido / Desconhecido
Completamente	Multiagente	Determinístico	Sequencial	Estático	Discreto	Conhecido

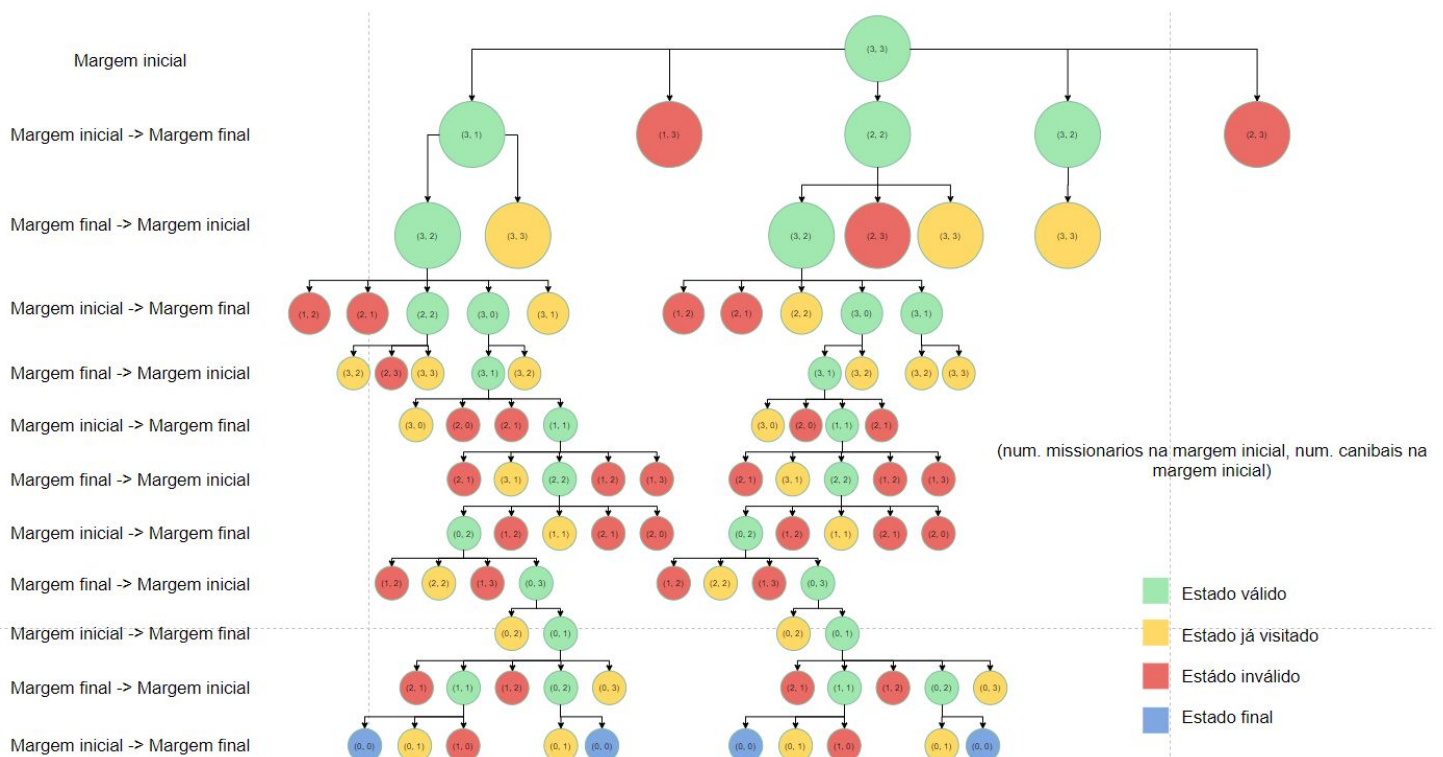
3. Diagrama do espaço de estados completos

a.2) Faça um diagrama do espaço de estados completo.

Para gerar o espaço de estados, a fim de simplificação, às duas tuplas foram abstraídas para uma única tupla que representa o número de missionários na margem inicial e o número de canibais na margem inicial.

O indicativo à esquerda representa o sentido da travessia em um dado nível de profundidade da árvore de estados.

DIAGRAMA 1 - Diagrama do espaço de estados completos.



4. Algoritmo de Aprofundamento Iterativo

b.1) Implemente e resolva o problema de forma ótima, utilizando um algoritmo de busca apropriado (Aprofundamento Iterativo ou Bidirecional).

Como dito na introdução, este problema foi resolvido através de um algoritmo de busca sem informação que utiliza o mecanismo de aprofundamento iterativo.

Formalmente, um algoritmo de busca em profundidade realiza uma busca sem informação que progride através da expansão do primeiro nó filho da árvore de busca, e se aprofunda cada vez mais, até que o alvo da busca seja encontrado ou até que ele se depare com um nó que não possui filhos (nó folha). Então a busca retrocede (backtrack) e começa no próximo nó.

A complexidade espacial de um algoritmo de busca em profundidade é muito menor que a de um algoritmo de busca em largura. A complexidade temporal de ambos algoritmos são proporcionais ao número de vértices somados ao número de arestas dos grafos aos quais eles atravessam.

Quando ocorrem buscas em grafos muito grandes, que não podem ser armazenadas completamente na memória, a busca em profundidade não termina, em casos onde o comprimento de um caminho numa árvore de busca é infinito. O simples artifício de “lembrar quais nós já foram visitados” não funciona, porque pode não haver memória suficiente.

Neste sentido, se faz útil o algoritmo de aprofundamento iterativo, no qual estabelece um limite de profundidade a ser expandida na árvore, e expande-se este limite iterativamente até encontrar um estado objetivo.

Para este problema, o espaço de estados foi tratado como uma árvore de busca, onde aplicou-se o algoritmo a partir do nó raiz (estado inicial), até encontrar um nó objetivo, incrementando a cada iteração, o limite de profundidade da busca.

5. Estados Repetidos

b.2) É uma boa ideia verificar a existência de estados repetidos?

A ideia de verificar a existência de estados repetidos é boa, pois reconhecendo que o estado já foi visitado, pode removê-lo, pois será uma busca repetida, reduzindo o espaço de busca no grafo de estados, consequentemente diminuindo o tempo para achar a solução ótima.

6. Dificuldades

c) Por que você imagina que as pessoas têm dificuldades para resolver esse quebra-cabeça, considerando que o espaço de estados é tão simples?

Mesmo tendo o espaço de estados simples, o problema não é trivial e muitas pessoas têm dificuldades para resolvê-lo. Isso se dá devido a ampla gama de restrições. Qualquer ação pode resultar em um estado inviável e mesmo que o problema possua um pequeno espaço de estados, estes estados são bem específicos (devido às restrições mencionadas anteriormente).

7. Conclusões

Foi possível observar, durante a realização do trabalho, a importância de realizar um levantamento preciso das especificações necessárias para assegurar uma solução válida, bem como a escolha correta do algoritmo e das decisões de implementações a serem tomadas.

8. Referências Bibliográficas

Russell, Stuart, and Peter Norvig. "Artificial intelligence: a modern approach." (2002).