

Relatório Agentes de resolução de problemas

Daniel Rosendo de Souza

¹IFCE– Instituto Federal do Ceará – Campus Maracanaú

Resumo. *Agentes de resolução de problemas são agentes baseados em objetivos, ou seja, consideram ações futuras e a conveniência de seus resultados para a obtenção de sucesso. Neste relatório iremos implementar agentes de resolução de problemas por meio das buscas A* e gulosa pela melhor escolha, para solucionar problemas clássicos como o do Aspirador de pó e o Mapa da Romênia.*

1. Introdução

Agentes reativos as vezes se deparam com ambientes com variáveis e regras condição-ação cujos números são grandes demais para serem armazenados. Neste caso é possível que seja construído um agente baseado em objetivos chamado de Agente de resolução de problemas, que diante das mais diversas possibilidades pode decidir o que fazer comparando diferentes sequências de ações possíveis. Esse processo é conhecido como busca, uma vez encontrada a solução, o agente pode executar a sequência de ações para chegar no objetivo.

O método da busca é dividido em três tipos: Busca Cega, Busca Heurística e Busca Local. Na busca Cega, não se sabe qual o melhor nó a ser visitado para que chegue ao estado objetivo, apenas distinguindo este do estado não objetivo. Na busca Heurística, é estimado qual o melhor nó a ser visitado com base em funções heurísticas. E na busca local é operado apenas em um único estado e o caminho move-se através da vizinhança desse estado. Todas as estratégias de busca se distinguem pela ordem em que os nós são expandidos.

Neste trabalho iremos abordar a Busca Heurística, mais especificamente os algoritmos de busca A* e gulosa pela melhor escolha para solucionar o problema do Aspirador de pó e do Mapa da Romênia

2. Embasamento Teórico

1. Busca

A busca heurística utiliza conhecimento específico sobre o problema para encontrar soluções de forma mais eficiente que a busca cega. A forma de busca A* e gulosa por melhor escolha são exemplos de busca heurísticas, e iremos estudá-las para a realização do trabalho.

A busca gulosa pela melhor escolha expande o nó que parece mais próximo ao objetivo de acordo com sua função heurística, como uma forma gulosa, como sugerido pelo seu nome. Não é ótima pois segue o melhor passo, considerando somente o caminho atual. Podendo haver um caminho melhor seguindo algumas opções piores

em alguns pontos da árvore de busca. Esta forma estar suscetível a criação de loops pois a minimização da função heurística é suscetível a falsos inícios.

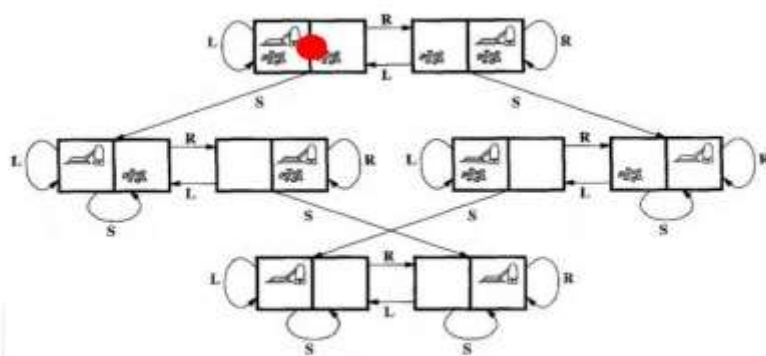
A busca A* tem como ideia principal evitar a expansão de caminhos que sejam caros, calculando o custo atual do caminho + o custo estimado para chegar ao objetivo. A busca A* degenera para a busca em amplitude quando a heurística não aponta preferência por nenhum caminho ao longo da busca. Ela é completa, ótima e eficiente.

2. Problemas

As problemáticas abordadas para testar a utilização dos algoritmos citados na sessão anterior será o aspirador de pó e o mapa da Romênia.

- O Aspirador de pó é considerado um mini problema, ou seja, pode ter uma descrição concisa e exata – pode ser utilizado com facilidade por diferentes sistemas de busca, com a finalidade de comparar o desempenho de algoritmos.

O agente está em uma entre duas posições, cada uma das quais pode conter sujeira ou não, o objetivo é manter ambos os ambientes limpos. Qualquer estado pode ser considerado inicial e sua função sucessora permite que seja gerado estados válidos resultantes de três ações: Mover-se para direita, mover-se para esquerda e limpar.

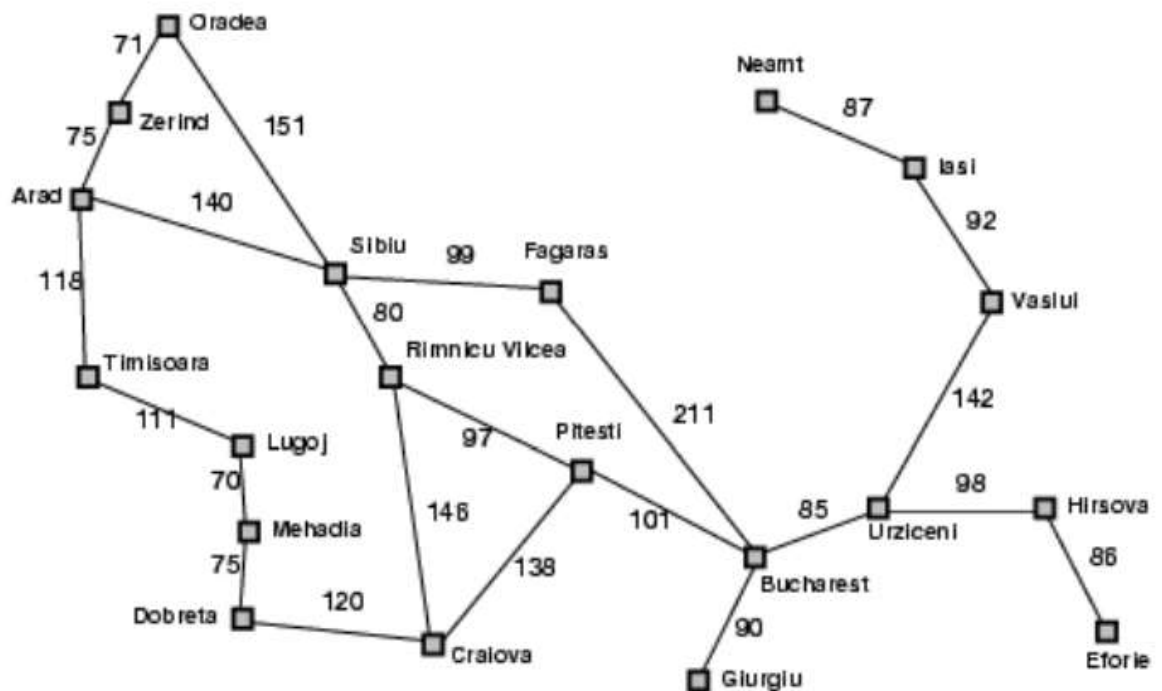


Os arcos denotam ações: E = Esquerda, D = Direita, A = Aspirar

- O mapa da Romênia, é considerado um problema de mundo real, pois tendem a não apresentar uma única descrição consensual, mas é possível fornecer uma ideia geral de suas formulações.

É composto por um mapa que expõe algumas cidades da Romênia e o custo para se locomover entre cada uma delas. O problema é encontrar o menor caminho, ou seja, que tenha o menor custo, partindo de uma cidade origem até

a cidade destino.



3. Implementação

A implementação deste problema, foi feita utilizando a linguagem Python na sua versão 2.7, utilizando como base o trabalho passado, onde implementamos os agentes por meio de busca, utilizando outros algoritmos, fizemos algumas modificações em referência ao projeto passado, a implementação dos algoritmos A estrela e Busca Gulosa pela melhor escolha, adicionando também as heurísticas de cada problema, mapa da Romênia e mundo do aspirador de pó.

A heurística do mapa da Romênia já era pré-definida pela literatura, que consistia em a distância em linha reta de Bucareste, entretanto no conceito base do mundo do aspirador de pó, não se é considerado custos de cada ação para os atos do aspirador e nem uma heurística para ele. Em nosso projeto, adicionamos custo de 1 para cada movimento do aspirador de pó entre os estados, e em sua heurística custo 2 para cada ação de limpar os dois cômodos, 1 para cada ação de apenas limpar um cômodo e custo 0 quando os dois estados estão limpos.

Como teste de objetivo para o mundo do aspirador de pó basta apenas estar com os dois estados limpos e para o mundo do aspirador de pó chegar ao seu destino final.

4. Resultados Obtidos

Em relação as implementações anteriores que não levavam em consideração o custo entre as ações, os algoritmos atualmente implementados vão desenvolver o seu caminho em base em um custo de deslocamento entre as ações e uma heurística para cada ação.

Para o mundo do aspirador de pó, não é passado um objetivo, pois ele já tem como base que ele possui apenas um objetivo único, que é deixar os dois estados limpos, então no nosso algoritmo só passamos o nosso estado inicial do problema, e ele se encarregará em achar uma solução ou direita limpo limpo ou esquerda limpo limpo e esse é o nosso teste de objetivo para o mundo do aspirador de pó. Para o mapa da Romênia, poderíamos ter diversos objetivos a serem alcançados, porém como possuímos apenas a heurística da distancia em linha reta da cidade de Bucareste, independente do nosso início, ele é apenas válido para um objetivo único que é Bucareste, qualquer outro ele apresentará uma tela de erro.

Para termos como base uma visualização inicial e simplista do que foi dito, será mostrado inicialmente o problema do mapa da Romênia, mostrando a diferença de caminhos e custo dos algoritmos. Inicialmente vamos colocar o teste que utilizamos em sala de aula, nosso ponto de partida é Arad com objetivo de chegada em Bucareste. Rodando o nosso algoritmo obtemos o seguinte resultado:

Busca A*	Busca Gulosa Pela Melhor Escolha
(['Arad', 'Sibiu', 'Rimnicu', 'Pitesti', 'Bucharest'], 'Custo do Caminho: 418')	(['Arad', 'Sibiu', 'Fagaras', 'Bucharest'], 'Custo do Caminho: 450')

Podemos notar que pela busca gulosa obtivemos o caminho com um custo acima da busca A, porém a quantidade de cidades (estados) visitados é menor que a busca A*, isso demonstra que nem sempre visitar menos estados, significa menos gasto energético, ou custo de caminho.

Rodando outra bateria de testes, agora tendo como ponto de partida Timensoara e como objetivo Bucharest, temos como resultado:

Busca A*	Busca Gulosa Pela Melhor Escolha
(['Timisoara', 'Arad', 'Sibiu', 'Rimnicu', 'Pitesti', 'Bucharest'], 'Custo do Caminho: 536')	(['Timisoara', 'Lugoj', 'Mehadia', 'Drobeta', 'Craiova', 'Pitesti', 'Bucharest'], 'Custo do

	Caminho: 615')
--	----------------

Novamente é notado a diferença entre caminhos e custo entre o deslocamento, busca pela melhor escolha, agora ocorreu o inverso, a busca A* andou por menos estados possuindo um custo menor que a busca gulosa, e a busca gulosa não foi tão eficiente assim, passando por mais estados e tendo um custo maior.

Exemplificando como objetivo outra cidade sem ser Bucareste a seguinte mensagem é nos mostrada:

Objetivo Não é Bucharest - Heuristica Somente Linha Reta Bucharest

Abaixo uma tabela comparativa entre os algoritmos, agora também tendo como base o mundo do aspirador de pó

	Busca A*	Busca Gulosa
Arad -> Bucharest	(['Arad', 'Sibiu', 'Rimnicu', 'Pitesti', 'Bucharest'], 'Custo do Caminho: 418')	(['Arad', 'Sibiu', 'Fagaras', 'Bucharest'], 'Custo do Caminho: 450')
Timisoara -> Bucharest	(['Timisoara', 'Arad', 'Sibiu', 'Rimnicu', 'Pitesti', 'Bucharest'], 'Custo do Caminho: 536')	(['Timisoara', 'Lugoj', 'Mehadia', 'Drobeta', 'Craiova', 'Pitesti', 'Bucharest'], 'Custo do Caminho: 615')
Origem ESS	(['ESS', 'ELS', 'DLS', 'DLL'], 'Custo do Caminho: 3')	(['ESS', 'ELS', 'DLS', 'DLL'], 'Custo do Caminho: 3')
Origem DSS	(['DSS', 'DSL', 'ESL', 'ELL'], 'Custo do Caminho: 3')	(['DSS', 'DSL', 'ESL', 'ELL'], 'Custo do Caminho: 3')
Origem ELS	(['ELS', 'DLS', 'DLL'], 'Custo do Caminho: 2')	(['ELS', 'DLS', 'DLL'], 'Custo do Caminho: 2')
Origem DSL	(['DSL', 'ESL', 'ELL'], 'Custo do Caminho: 2')	(['DSL', 'ESL', 'ELL'], 'Custo do Caminho: 2')