Relatório Atividade 1: A* Para Caixeiro Viajante

Nome: Lucas de Araújo Torres e Francisco Pedro Mota Braga Carneiro
Matrícula: 557156, 501039
Professor: Samy Soares Passos de Sá

Resumo

Este relatório tem o objetivo de explicar e especificar o trabalho 2 da cadeira de Inteligência Artificial. O trabalho consiste em escrever o algoritmo do A* usando o problema do caixeiro viajante. Uma heurística que foi dada para calcular h(n) seria usar uma árvore geradora mínima, pois está bem explicada na descrição do trabalho.

Note que fizemos os códigos em inglês a fim de postar futuramente para trabalhos de portfólios.

Descrição da Atividade

Explicando a atividade, fizemos uma função que computa a árvore geradora mínima pelo algoritmo de Prim. Sentimos mais a vontade de implementar ele pois sabiamos que ele funciona muito bem. Implementamos uma função chamada AStarProcedure, que faz todo o procedimento de calcular o f(n) dado um grafo qualquer.

Sentimos mais a vontade usando dicts para representar um grafo, onde cada chave representa um vértice e seu valor uma lista de tuplas, onde cada tupla tem a distância e o vértice adjacente. Isso é chamado também de lista de adjacências.

Existe uma função chamada h_calc(), que calcula o custo de h(n) da heurística. A implementação do código dele mostra bem o que é retornado.

Usamos uma fila de prioridades que guarda sempre o menor valor a ser o primeiro a sair. É importante usar essa Ed, pois melhora na eficiência do código. Ela está tanto presente no algoritmo prim como na função AStarProcedure().

Todas as funções estão escritas em functions.py e o código principal com os testes está escrito na main.py. É ele que deve ser rodado.

Resultados

Para a primeira hipótese, criamos uma função que troca o primeiro nó a ser visitado e coloca ele no começo da lista. Essa foi uma forma de escolhermos o nó aleatoriamente sem precisar mexer no código de functions.py. Temos que nesse primeiro experimento, o custo deve se manter igual para qualquer escolha de nó inicial. Isso por que os pontos são os mesmos e a escolha do nó inicial não deve implicar nos resultados. Para grafos com mais de 10 pontos, será escolhidos 10 pontos aleatórios para serem os iniciais.

Sobre a segunda hipótese, não deu bem para elaborar bons testes no trabalho. Porém, fizemos algumas modificações enquanto implementava, e notamos que não alterava nos resultados finais a forma de como era incluído o 0 ou não. Dessa forma, não implicou em nada.

Conclusão

Concluímos nesse trabalho que foi um pouco mais difícil de implementar que a subida da encosta. Não conseguimos elaborar bem todos os testes e casos, mas vimos que o algoritmo é bom de ser usado pois ele sempre garante o resultado ótimo, se for usado uma heurística admissível, é claro.