

Resolução de Problemas - Algoritmos de Procura

Instrumento de Avaliação em Grupo

Inteligência Artificial, 3º ano, 1º semestre, 2024/2025

André Barbosa Teixeira - a104002
João Andrade Costa - a104258

Diogo Alexandre Pereira da Costa - a95147
Nuno Martins - a96227

Grupo 37



Avaliação pelos pares

A104002 André Barbosa Teixeira DELTA = 0.2

A95147 Diogo Alexandre Pereira da Costa DELTA = 0.2

A104258 João Andrade Costa DELTA = 0.2

A96227 Nuno Martins DELTA = -0.6



Índice

Avaliação pelos pares.....	2
Índice.....	3
Introdução: Descrição do problema.....	4
Formulação do problema.....	4
Tipo de problema:.....	4
Estado inicial:.....	4
Atributos do nodo:.....	4
Atributos do veículo:.....	5
Atributos do mapa:.....	5
Teste objetivo:.....	5
Operadores:.....	5
Custo da solução:.....	5
Design e Implementação.....	6
Comparação das estratégias.....	8
Decisões do grupo.....	8
Conclusão.....	9



Introdução: Descrição do problema

Durante uma catástrofe natural, a necessidade de fornecer bens essenciais às zonas mais afetadas torna-se uma prioridade urgente. As áreas atingidas podem variar em termos de gravidade e necessidades, desde grandes centros populacionais a regiões mais isoladas e de difícil acesso. Cada uma dessas zonas apresenta desafios específicos, sendo que a sua acessibilidade depende de fatores como a geografia local e as condições meteorológicas em constante mudança. Para chegar a essas áreas de forma eficiente, são usadas várias formas de transporte (camiões, helicópteros, comboios, e barcos), sendo que cada veículo possui as suas limitações próprias: capacidade de carga, rotas disponíveis, e eficiência energética. A seleção dos veículos apropriados para cada entrega e otimização da rota são fundamentais para garantir que os mantimentos cheguem a tempo, especialmente em zonas críticas.

A tarefa principal é desenvolver uma estratégia que maximize a cobertura das zonas afetadas, minimizando o custo de operação dos veículos. Para atender várias localizações com diferentes carências, é necessário um sistema de prioridades que nos informa a ordem a seguir para otimizar a distribuição e certificar que as zonas críticas são tratadas primeiro, salvando o máximo número de vidas possível.

Formulação do problema

Tipo de problema:

Problema de contingência, uma vez que o ambiente não é determinístico, sendo que vários eventos podem causar alterações no estado, sendo necessário intercalar procura e execução.

Estado inicial:

Ponto de início no nodo representante da cidade 1 com carga carregada suficiente para atender todas as necessidades atuais.

Atributos do nodo:

- Nome da cidade;
- Identificador (incrementado manualmente na população);
- Necessidade;
- Mantimentos atuais.



Atributos do veículo:

- Tipo (camião, combóio, helicóptero, barco);
- Velocidade;
- Capacidade de carga;
- Eficiência energética.

Atributos do mapa:

- Lista de nodos;
- Lista de nodos e respectivas conexões, incluindo custo, lista de acessibilidade e estado (acessível/inacessível);
- Lista de heurísticas para algoritmos informados.

Teste objetivo:

Percorrer todas as cidades na lista de prioridades, minimizando o custo das viagens.

Operadores:

- Escolher/trocar veículo;
- Mover para nodo adjacente;
- Carregar veículo;
- Descarregar veículo.

Custo da solução:

O custo da solução é uma função que multiplica a distância entre dois nodos adjacentes (valor pré-determinado na população do mapa) pelo consumo de um veículo, de seguida multiplica este valor pelo quociente entre a carga a transportar e a capacidade do veículo (uma vez que, se a carga exceder a capacidade do veículo, vão ser necessários mais para a transportar), arredondando para cima. Depois, passa este valor para metade e adiciona o quociente entre 500 e a velocidade do veículo (500 foi escolhido pois alterava de forma satisfatória o custo de acordo com a velocidade).

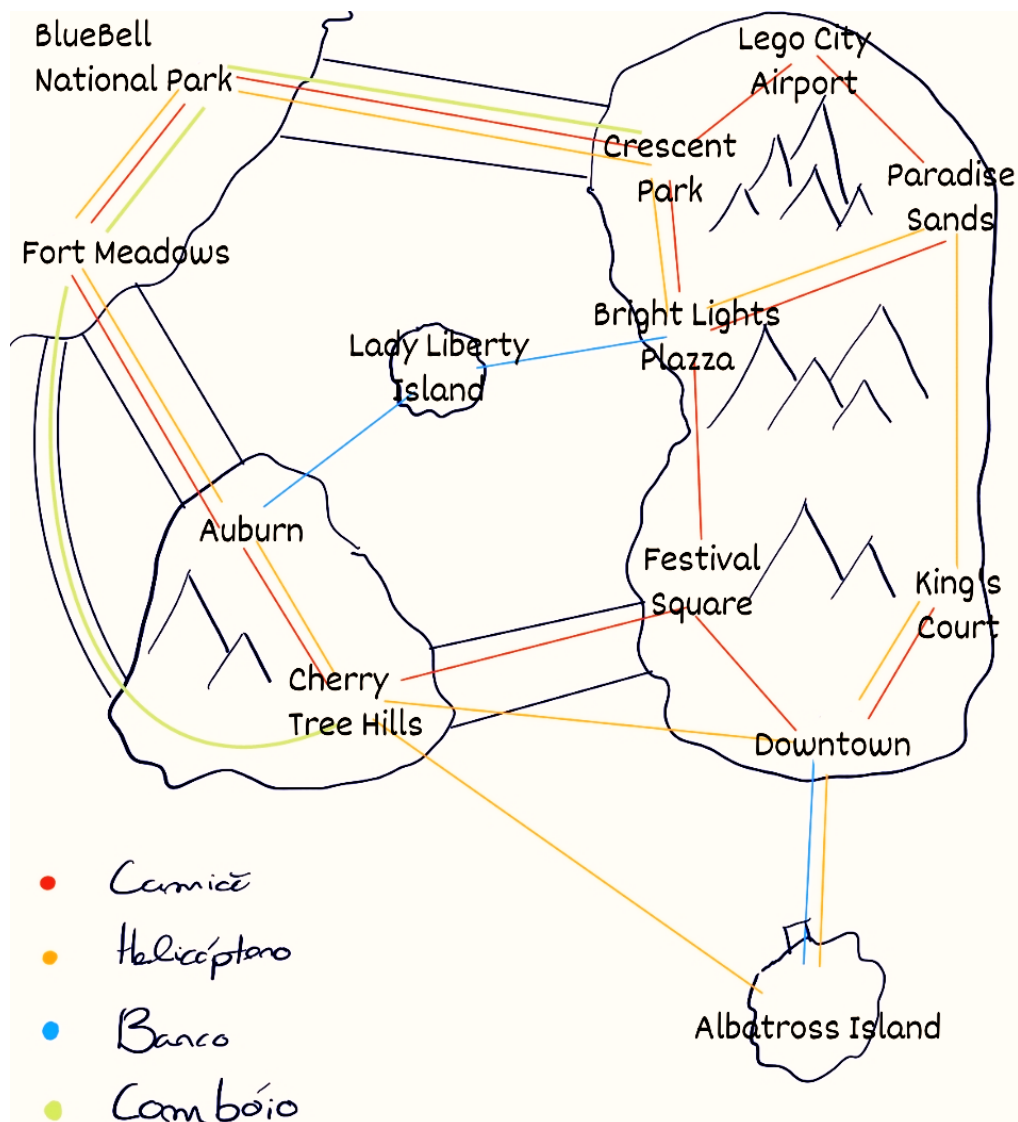
Por exemplo, para uma dada travessia, de carga a transportar c , o custo da transição de um nodo para outro adjacente é:

$$\text{custo} = 0.5 * ((\text{distância} * \text{consumo}) * \text{ceiling}(c/\text{capacidade})) + (500 / \text{velocidade})$$



Design e Implementação

Após a formulação do problema, trabalhamos de forma a poder passar a informação que tínhamos para a forma de um grafo, permitindo assim a aplicação de algoritmos de procura já estudados e implementados em fichas práticas. Formulamos então o grafo abaixo sendo cada cidade representada por um nodo construído pelo seu nome e as conexões entre si (as arestas do grafo) identificadas por cor, relativo ao meio de transporte que as pode atravessar.



Topografia do problema



Para preparar a execução dos algoritmos, são necessárias definir as cidades com necessidades após uma catástrofe, para isso o utilizador introduz cada escolha no programa bem como o número de carga que esta necessita e cada cidade é colocada numa lista de prioridades e ordenada com base na sua necessidade: quanto maior for, mais à frente fica.

Antes de serem feitas as travessias, o nodo inicial escolhido é o 1, com o nodo final o que se apresenta à frente da lista de prioridades, sendo que os outros nodos nessa lista levarão “incentivo” de escolha para o algoritmo passar por eles a caminho da cidade com maior necessidade para passar otimizar a rota escolhida. O incentivo consiste em baixar o custo para um nodo na lista por uma quantidade fixa dependendo da sua posição nesta mesma. Por exemplo, se um item estiver no início da lista de prioridades, o seu custo final passará a ser 70% do custo atual calculado.

Para prosseguir com as travessias, mesmo os algoritmos não informados não se podem limitar a escolher as conexões numa determinada ordem, uma vez que cada conexão tem 1 ou mais meios de transporte possíveis, logo, foi implementada uma função que calcula o custo de cada uma dessas opções ([vista acima](#)) e devolve o menor custo encontrado, bem como o meio de transporte que o alcança. Para além disso, os algoritmos têm de verificar se cada rota se encontra acessível, pois pode mudar de acordo com as condições meteorológicas ou eventuais bloqueios, sendo a rota ótima, de uma ponto para outro, variável. Para isso foram implementadas funções que mudam as rotas acessíveis para simular a mudança de condições.

Com isto podem então ser implementados os algoritmos de procura, sendo estes DFS e BFS para procura não informada e A* para procura informada. Porém, em vez de devolverem apenas um caminho simples, devolvem uma lista em que cada elemento consiste de ambos os nodos de cada aresta percorrida e o veículo utilizado para a percorrer, isto uma vez que a função de custo mínimo devolve o veículo utilizado. Para além disso, devolvem também o custo total da solução. Como para a pesquisa informada são necessárias heurísticas para auxiliar na procura, é mantida uma lista destas, com uma função para definir os seus valores. Estas heurísticas são determinadas a cada iteração do algoritmo A* pela diferença entre o número identificador do nodo atual e o número identificador dos outros nodos, sendo um método eficaz uma vez que cidades com números próximos também se encontram próximas.

Após uma execução de qualquer algoritmo, se ainda existirem cidades com necessidades não atendidas, recomeça o algoritmo com nodo inicial o destino da travessia anterior e o novo destino o primeiro elemento da lista de prioridades, sendo este removido da mesma. Em contrapartida, cada vez que uma cidade na lista for atravessada, ou o algoritmo chegar ao seu destino, a carga é decrementada no valor igual à sua necessidade.



Comparação das estratégias

Após a implementação das 3 estratégias de procura, observamos que, independentemente do cenário, DFS tem de longe a pior performance, obtendo custos acumulados duas vezes maiores que os outros algoritmos. De facto, quando se simulam obstruções em caminhos, este conseguia várias vezes reduzir o seu custo, uma vez que o número de rotas a explorar é bem reduzido.

A* e BFS têm performance muito semelhantes, porém, quando num cenário sem restrições, A* é em média 12.5% mais eficiente que BFS, porém, quando os caminhos são restringidos, a diferença dos dois torna-se negligenciável.

Decisões do grupo

Face ao problema apresentado, tomamos algumas decisões onde baseamos a nossa resolução. A primeira é que não contabilizamos o reabastecimento dos veículos, apenas somamos o custo total da travessia, sendo esta interpretada como um planeamento de rota de uma situação real antes de esta ser executada e não uma implementação da mesma sem preparação prévia. Para além disso permitimos a troca de veículo atual para qualquer deslocamento, isto uma vez que há áreas que são só acessíveis por 1 veículo a partir de um nodo adjacente, que teremos de chegar utilizando outros veículo, ou seja, para evitar bloqueios, a troca é sempre permitida, tendo como objetivo determinar o melhor tipo de veículo para cada deslocação.

Para cada entrega, decidimos que a carga é genérica, não sendo feita a distinção entre os tipos de mantimentos a entregar, sendo apenas entregue a quantidade especificada a cada nodo. Para além do mais, no início de cada entrega é carregada carga o suficiente para satisfazer todos os pedidos na lista de prioridades, isto é, se for selecionada a operação em forma automática, caso o utilizador deseje, este pode escolher a carga a inserir numa travessia, sendo esta depois descarregada nos nodos que a necessitam e, se não for suficiente, indica onde falhou, e o utilizador terá de continuar até as cidades estarem todas atendidas. Por fim, tempo de entrega não é contabilizado no resultado, porém, como podemos observar pelo cálculo de custo, influencia o resultado, sendo que a velocidade de cada veículo impacta este mesmo, garantindo que, para além de custos reais serem minimizados, o método escolhido também é o que permite fazer isso em menos tempo.



Conclusão

Neste trabalho, foi-nos apresentado um problema real. Este tratava-se de um problema de contingência ao qual pudemos aplicar algoritmos de procura informada e não informada para obter várias soluções com diferentes níveis de desempenho. Implementamos os algoritmos *DFS*, *BFS* e *A**, que foram lecionados nas aulas e tiveram de ser adaptados ao caso de uso do nosso problema.

O desempenho dos algoritmos foi tal como esperado: O *DFS* teve os piores resultados devido ao largo número de cidades, seguido pelo *BFS* que tenta minimizar o número de movimentos e a *A** obteve os melhores resultados devido à sua estratégia de procura informada.

Este trabalho proporcionou a aplicação dos conceitos lecionados nas aulas e não só consolidou a matéria como nos ensinou a aplicar os algoritmos de procura a situações reais com cenários dinâmicos e complexos.