Universidade Federal Rural de Pernambuco

Departamento de Estatística e Informática

Bacharelado em Sistemas de Informação

FLYFOOD

Italo de Lima Silva

Recife

Fevereiro de 2023

1. Introdução

1.1 Apresentação e Motivação

No cenário fictício de um futuro próximo, as empresas que fornecem serviço de delivery precisam de uma nova alternativa para que possam continuar fazendo seu trabalho de maneira efetiva, devido à supervalorização da mão-de-obra humana. Dessa forma, a alta demanda de clientes pode ser atendida, e consequentemente, o seu tempo de espera é reduzido.

A partir dessa problemática, surge o FlyFood, uma espécie de delivery aéreo caracterizado pelo uso de drones para a realização de entregas nos locais onde o serviço é solicitado.

Para fazer as entregas, cada drone recebe uma rota, passa pelos pontos de entrega e volta ao restaurante. No entanto, o trajeto feito não é percorrido a fim de otimizar o tempo da entrega, pois ele não avalia o melhor caminho a ser seguido, bem como não seleciona a ordem dos pontos de entrega. E além disso, o limite de duração da bateria dos drones também é um empecilho para que todos os pedidos sejam atendidos em tempo hábil.

A situação apresentada, apesar de hipotética, retrata um contexto não muito difícil de se concretizar na sociedade atual, visto que, com o constante avanço tecnológico das máquinas, a mão-de-obra humana se torna mais ameaçada a cada dia. Pois, se hoje existem tantos algoritmos, softwares e inteligências artificiais capazes de realizar tarefas difíceis de forma otimizada e eficaz, verifica-se intuitivamente, que alternativas como o FlyFood fazem parte do planejamento de um futuro tangível e inovador. Por essa razão, este trabalho visa sanar as falhas que esse projeto promissor apresenta, através de um “algoritmo de força bruta” que será desenvolvido no decorrer destas páginas.

1.2 Formulação do problema

Matriz de entrada:

```
6 6
0 0 0 B 0 0
0 E 0 0 0 0
R 0 0 0 0 C
0 D 0 0 0 0
A 0 0 0 G 0
0 0 F 0 0 0
```

Coordenadas(i, j) = (6, 6)

Ponto de partida/chegada = {'R': (2, 0)}

Pontos de entrega = {'B': (0, 3), 'E': (1, 1), 'C': (2, 5), 'D': (3, 1), 'A': (4, 0), 'G': (4, 4), 'F': (5, 2)}

Formulação matemática (Permutação):

$$n = n - 1!$$

(Gera todas as rotas possíveis)

Formulação matemática (Menor rota possível):

$$\min Z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\begin{aligned}
\text{sa} \quad & \sum_{i=1}^{n} x_{ij} = 1, && j = 1, 2, \ldots, n \\
& \sum_{j=1}^{n} x_{ij} = 1, && i = 1, 2, \ldots, n \\
& u_i - u_j + n x_{ij} \leq n - 1 && \forall 2 \leq i \neq j \leq n \\
& x_{ij} \in \{0,1\} && \forall i, j
\end{aligned}$$

1.3 Objetivos

Nesse viés, evidencia-se que o objetivo geral deste trabalho é propor uma solução que possibilite a otimização do tempo e da vida útil das baterias dos drones. Essa solução se baseia na implementação de um algoritmo de roteamento que deve avaliar todos os trajetos possíveis a serem realizados, e em seguida, revelar qual deles é o melhor a ser seguido, ou seja, o menor trajeto que atende a todas as entregas.

Mas, para que esse algoritmo funcione de forma precisa e eficiente, é preciso ter ciência de onde está o ponto de partida, que também será o local de retorno dos drones, assim como também precisamos identificar cada ponto de entrega, encontrar todas as rotas possíveis, calcular quantos “dronômetros” serão percorridos em cada uma, e finalmente, identificar a menor rota entre elas.

1.4 Organização do trabalho

A estrutura deste trabalho segue um padrão de ordem que visa trazer uma melhor compreensão acerca da elaboração do projeto, envolvendo também os conceitos

matemáticos fundamentais para o desenvolvimento do algoritmo proposto. Sendo assim, a introdução ilustra uma visão geral da problemática a ser resolvida e evidencia a relevância e o objetivo do projeto; já o referencial teórico esclarece os conceitos que levaram a elaboração da solução, bem como os métodos pré-existentes nos quais o FlyFood se baseia.

Em seguida, a sessão de trabalhos relacionados traz mais algumas informações através de projetos similares ao FlyFood; A metodologia do trabalho é constituída por um pseudocódigo do projeto, e os testes realizados durante a elaboração do algoritmo estão à mostra na parte dos experimentos; e por fim, o resultado e a conclusão detalham a resolução do problema narrado na introdução, mostrando como as fraquezas do projeto inicial foram tratadas pelo algoritmo.

2. Referencial Teórico

Nesta seção, veremos como o clássico problema do caixeiro viajante foi adaptado para o algoritmo do atual projeto. Também será feita a análise da complexidade do algoritmo, para determinar a classe do problema.

2.1 Problema do Caixeiro Viajante

The Travelling Salesman Problem-TSP

O problema do caixeiro viajante é amplamente utilizado como exemplo na matemática, pois abrange não só a otimização combinatória, como também grafos e problemas computacionais. Sendo assim, dado um conjunto de cidades dentro de uma matriz, o TSP consiste em encontrar uma rota que:

- Parta de uma cidade origem e retorne para ela no fim do percurso;

- Passe em cada uma das outras cidades apenas uma vez;

- Percorra uma rota com a menor distância possível.

A partir da análise dessas informações acerca do problema do caixeiro viajante, verifica-se que a solução ótima a ser encontrada nesse problema é a mesma que está sendo elaborada no FlyFood. Com isso, conclui-se que o TSP pode ser utilizado como um facilitador no processo de construção do algoritmo desejado no projeto.

Para uma melhor compreensão desse problema, ele pode ser modelado via grafo:

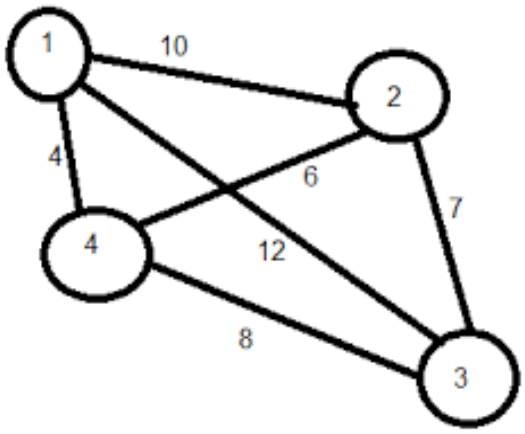


Figura 1. Exemplo de grafo com ciclo hamiltoniano

Tendo em vista que um ciclo hamiltoniano em um grafo é um caminho que começa em um vértice, percorre os outros vértices uma única vez e retorna à origem, podemos concluir que o grafo acima representa um problema do caixeiro viajante, que, por conseguinte, também faz referência à ideia do FlyFood. Nesse caso, a solução por força bruta desse grafo é: Encontrar todos os ciclos hamiltonianos do grafo e identificar qual deles tem o menor custo (menor distância).

Por fim, sabendo que um grafo completo com n vértices possui (n-1)! ciclos hamiltonianos, a tese apresentada na formulação matemática por permutação na introdução agora pode ser comprovada através da aplicação do problema nos grafos.

2.2 Classe do problema

No contexto da teoria da complexidade computacional, o problema do caixeiro viajante é caracterizado como um problema de classe NP-Difícil, pois é notável que não existe um algoritmo eficiente para resolvê-lo.

O fato de estarmos lidando com um problema de otimização, e não com problemas de decisão ou de pesquisa, não influencia na classe do problema, pois existe uma dificuldade intrínseca ao problema, que não o permite ter um algoritmo ótimo.

3. Trabalhos relacionados

3.1 Roteamento de leituristas: um problema np-difícil

Apesar do problema abordado nesse artigo abranger uma área totalmente diferente do FlyFood, a complexidade de ambos é a mesma. Através da comparação desses 2 artigos, é possível compreender o roteamento como um algoritmo de vasta aplicabilidade.

USBERTI, Fábio Luiz. ROTEAMENTO DE LEITURISTAS: UM PROBLEMA NP-DIFÍCIL. Disponível em: http://www.din.uem.br/sbpo/sbpo2008/pdf/arq0020.pdf

3.2 Ciclos hamiltonianos em grafos Kneser

Já foi visto que os ciclos hamiltonianos são uma ferramenta imprescindível para a identificação de grafos representantes do problema do caixeiro viajante, que é a base do algoritmo que foi construído. Neste artigo, os ciclos hamiltonianos são inseridos como objeto de estudo dentro de grafos Kneser, o que traz uma complexidade a mais para esse elemento.

BUENO, Letícia Rodrigues. Ciclos hamiltonianos em grafos Kneser. Disponível em: https://www.pesc.coppe.ufrj.br/uploadfile/1263297012.pdf

3.3 Experimentos computacionais com heurísticas de melhorias para o problema do caixeiro viajante

O objetivo deste trabalho é analisar aspectos específicos da implementação computacional de heurísticas de melhorias. Algo relevante para problemas que não possuem uma solução totalmente eficiente, como o FlyFood.

CUNHA, Claudio Barbieri da. EXPERIMENTOS COMPUTACIONAIS COM HEURÍSTICAS DE MELHORIAS PARA O PROBLEMA DO CAIXEIRO VIAJANTE. Disponível em: https://www.researchgate.net/profile/Claudio-Cunha-3/publication/228434832_Experimentos_computacionais_com_heuristicas_de_melhorias_para_o_problema_do_caixeiro_viajante/links/54803ccb0cf2ccc7f8bb2c18/Experimentos-computacionais-com-heuristicas-de-melhorias-para-o-problema-do-caixeiro-viajante.pdf

4. Metodologia

Na elaboração do algoritmo de força bruta, existem 3 etapas cruciais para o funcionamento eficiente do mesmo, que são:

4.1 Leitura da matriz

Antes de qualquer coisa, é necessário buscar alternativas para acessar a raiz do problema, que é o documento com a matriz de roteamento. A partir disso, os dados precisam começar a serem tratados, fazendo o programa reconhecer os elementos que se referem a coordenadas, pontos de entrega, ponto de origem e etc. Dessa forma, com os dados da matriz passados como valores de uma nova função, o algoritmo pode começar a surgir.

4.2 Permutação de Rotas

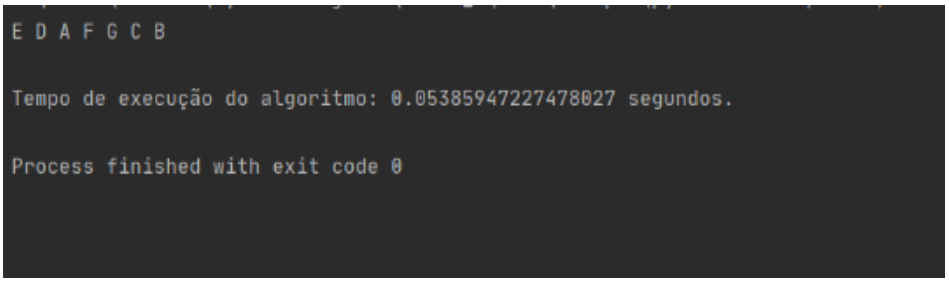
Contendo as informações necessárias para o início do algoritmo, neste momento, é realizado o cálculo de todas as rotas possíveis que o algoritmo pode oferecer, isso é feito através dos valores das coordenadas e das linhas e colunas da matriz. Consequentemente, o programa vai retornar todos os caminhos que podem serem percorridos.

4.3 Menor trajeto

Dentro de todas as possibilidades de rotas existentes, agora o programa será “forçado” a imprimir apenas uma possibilidade, que é aquela com menor custo de dronômetros.

5. Resultados

Através do algoritmo implementado, podemos encontrar a solução desejada na introdução deste trabalho: uma cadeia de caracteres contendo a ordem dos sete pontos de entrega que os drones seguirão:



6. Conclusão

Tendo em vista que o objetivo principal da construção do algoritmo era descobrir a menor rota possível para a preservação da bateria dos drones e para o lucro das empresas, pode-se concluir que esse objetivo foi concluído, isso se evidencia no resultado da seção anterior.

É notório que, além do programa responder da forma esperada, o algoritmo também demonstra ser dinâmico, pois se qualquer dado no arquivo da matriz for alterado, o algoritmo é atualizado para calcular uma nova rota a partir dessa “nova” matriz.

Sendo assim, a idealização do delivery de drones pode avançar para novos rumos, como por exemplo, um algoritmo que siga uma heurística, ao invés da força bruta, e dessa forma, o projeto pode crescer até uma possível implementação na vida real.

Referências Bibliográficas

Serpa, Matheus da, S. et al. Análise de Algoritmos. Disponível em: Minha Biblioteca, Grupo A, 2021.

Cormen, Thomas. Desmistificando Algoritmos. Disponível em: Minha Biblioteca, Grupo GEN, 2013.

Dobrushkin, Vladimir A. Métodos para Análise de Algoritmos. Disponível em: Minha Biblioteca, Grupo GEN, 2012.