Problema do caixeiro viajante aplicado ao contexto da caça ao pokémon na região da Universidade Federal do Piauí

Paulo E. R. Araujo[1], **Carlos M. Sousa**[1], **Gabriel R. Nunes**[1]

[1]Departamento de Computação - Universidade Federal do Piauí (UFPI)
Teresina - PI

pumbadeveloper@gmail.com

Abstract. *This paper presents a comprehensive study of the traveling salesman problem, including its definition, several solution approaches with their respective complexities, and its application in a real context. The study is based on the region of the Federal University of Piaui, which serves as a topological scenario for analysis and experimentation. The problem arises from the vendors' need to make deliveries in multiple locations, covering the shortest possible distance. In order to address this issue, we explore different solution strategies, some of them being the main focus of this work.*

Resumo. *Este artigo apresenta um estudo abrangente do problema do caixeiro viajante, incluindo sua definição, diversas abordagens de solução com suas respectivas complexidades e sua aplicação em um contexto real. O estudo é baseado na região da Universidade Federal do Piauí, que serve como cenário topológico para análise e experimentação. O problema surge da necessidade dos vendedores em efetuar entregas em múltiplos locais, percorrendo a menor distância possível. Com o intuito de abordar essa questão, exploramos diferentes estratégias de resolução, sendo algumas delas o foco principal deste trabalho.*

1. Introdução

O problema do caixeiro viajante é um desafio clássico da área de otimização combinatória, com ampla aplicação em diversas áreas, como logística, transporte, redes de distribuição, planejamento urbano e roteamento de veículos. Ele envolve encontrar o caminho mais curto que um viajante deve percorrer ao visitar um conjunto de cidades, passando por cada uma delas uma única vez e retornando ao ponto de partida [1]. Esse problema desperta grande interesse devido à sua natureza NP-completo, o que significa que não existe um algoritmo conhecido capaz de resolvê-lo de forma eficiente para instâncias de tamanho arbitrariamente grande.

O objetivo deste artigo é realizar um estudo aprofundado do problema do caixeiro viajante, explorando suas diferentes formulações, abordagens de solução e aplicações práticas. Para tanto, utilizaremos a região da Universidade Federal do Piauí como base topológica, com o intuito de fornecer um contexto real para a análise e experimentação.

O problema do caixeiro viajante pode ser formulado de maneira matemática como um problema de otimização combinatória. Dado um conjunto de cidades representado por um grafo completo, onde cada cidade é um vértice e as distâncias entre elas são representadas pelas arestas, o objetivo é encontrar um ciclo hamiltoniano de menor comprimento, ou seja, uma rota que visite todas as cidades exatamente uma vez, retornando à cidade de origem [2].

Formalmente, seja $G = (V, E)$ um grafo completo, onde V é o conjunto de vértices que representa as cidades e E é o conjunto de arestas que denota as distâncias entre as cidades. Seja d_{ij} a distância entre as cidades i e j. O problema do caixeiro viajante pode ser definido como encontrar a permutação das cidades $P = (v_1, v_2, ..., v_n)$ que minimize a função objetivo:

$$\min \sum_{i=1}^{n} (d_{v_i v_{i+1}}) + d_{v_n v_1}, \tag{1}$$

sujeito a:

- Cada cidade ser visitada exatamente uma vez: $v_i \neq v_j$ para $i \neq j, i, j \in \{1, 2, ..., n\}$, onde n é o número de cidades.
- Cada cidade, exceto a cidade de origem, ser visitada uma única vez: $\sum_{i=1}^{n}(v_i) = 2$ para $i \neq 1$, $i \in \{1, 2, ..., n\}$, onde $\sum_{i=1}^{n}(v_i)$ é a soma das ocorrências de cada cidade v_i no ciclo.

A função objetivo representa a soma das distâncias percorridas ao longo do ciclo hamiltoniano, onde $d_{v_i v_{i+1}}$ é a distância entre as cidades v_i e v_{i+1}, e $d_{v_n v_1}$ é a distância entre a última cidade visitada e a cidade de origem. As restrições garantem que cada cidade seja visitada exatamente uma vez, com exceção da cidade de origem, que é visitada duas vezes (no início e no fim do ciclo).

Assim, o problema do caixeiro viajante consiste em encontrar a permutação ótima das cidades que minimize a função objetivo, considerando todas as possíveis combinações de visitação, o que torna o problema desafiador devido ao crescimento exponencial do número de soluções possíveis à medida que o número de cidades aumenta.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta a aplicação do problema do caixeiro viajante no contexto da caça aos Pokémons, destacando a topologia extraída da região da Universidade Federal do Piauí que será utilizada como cenário para o estudo. Na Seção 3, é abordado a resolução do problema utilizando um algoritmo de força bruta e analisando a complexidade associada a essa abordagem. Por fim, na Seção 4, apresentaremos as conclusões obtidas a partir das análises realizadas, fornecendo uma visão abrangente do problema.

2. Aplicação do problema

No contexto deste estudo, uma aplicação intrigante do problema do caixeiro viajante surge na busca por Pokémon na região da Universidade Federal do Piauí. Com a popularidade do jogo Pokémon Go e pela presença de uma grande diversidade de Pokémons na área, a utilização do problema do caixeiro viajante é uma ótima maneira de otimizar a jornada dos caçadores. A região da universidade serve como cenário topológico, em que cada ponto de interesse é mapeado como um local potencial para encontrar Pokémon. Com o objetivo de maximizar a eficiência e minimizar a distância percorrida, o problema do caixeiro viajante se torna uma ferramenta essencial para planejar uma rota que passe por todos esses pontos, garantindo uma melhor experiência para os caçadores.

Para este estudo, foi realizado um processo de coleta de dados e definição dos elementos da topologia. Inicialmente, o mapa da Universidade Federal do Piauí (UFPI) foi extraído do Google Maps [3], fornecendo uma representação precisa da região em estudo (Figura 1).

Figura 1. Mapa da Universidade Federal do Piauí (UFPI) extraído do google maps.

Em seguida, como mostrado na Figura 2, foram identificados e definidos os pontos de interesse no mapa, que seriam considerados nós do problema do caixeiro viajante, sendo atribuída uma legenda a cada ponto para indicar sua representação. As arestas do mapa foram estabelecidas com base nas ruas e caminhos existentes, e as distâncias entre os pontos foram determinadas considerando variações físicas do percurso, obstáculos e características do terreno.

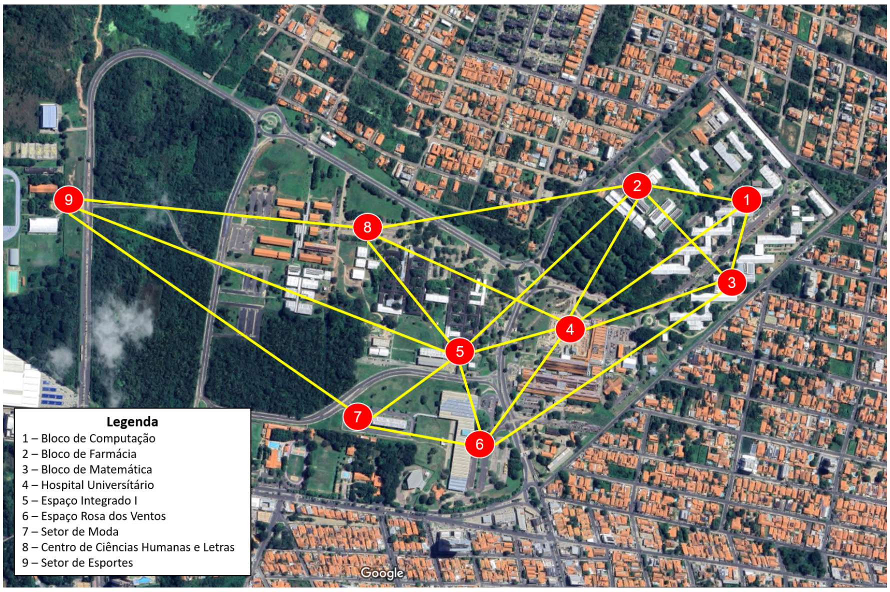

Figura 2. Pontos de capturas de pokémons e seus caminhos.

A topologia resultante, ilustrada na Figura 3, é composta por 9 nós e 19 arestas. Essa topologia será utilizada como entrada nos algoritmos apresentados neste estudo, permitindo a análise e resolução do problema do caixeiro viajante nesse contexto específico.

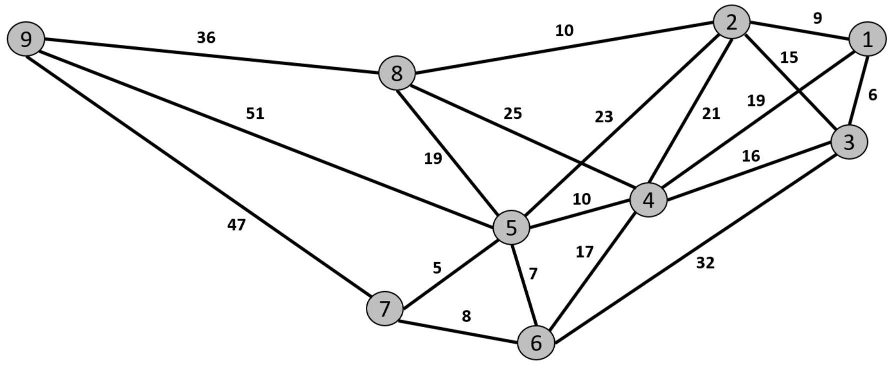

Figura 3. Topologia gerada a partir dos pontos de interesses.

3. Resolução por Força Bruta

A forma mais básica de resolução de qualquer problema é a aplicação da estratégia de força bruta, e isso não é diferente no caso do problema do caixeiro viajante. Este estudo explora a resolução desse desafio utilizando essa abordagem direta e exaustiva. A estratégia de força bruta consiste em testar todas as possíveis permutações de visitação dos pontos de interesse, calculando a distância total percorrida em cada iteração [4]. Embora seja uma solução exata, é importante ressaltar que sua eficiência diminui rapidamente à medida que o número de pontos aumenta, devido ao número de permutações possíveis crescer exponencialmente.

Na permutação, é calculado o número de maneiras diferentes que um conjunto de elementos pode ser organizado em uma ordem específica. O cálculo do número de permutações em um conjunto é feito usando a fórmula n! (n fatorial), onde n é o número de elementos no conjunto.

Para a resolução do problema, é criado um array contendo todos os nós do grafo, exceto o nó de partida, que será adicionado posteriormente. Em seguida, esse array é usado para gerar uma matriz com todas as permutações possíveis, onde cada linha representa uma permutação. Para formar um ciclo, o nó de partida é adicionado tanto no início quanto no fim de cada permutação. Dessa forma, obtemos uma matriz que abrange todas as rotas possíveis. Em seguida, o custo de cada rota é calculado com base nas distâncias entre os nós. Ao analisar os custos de todas as rotas, é realizada uma busca para encontrar a rota com o menor custo, indicando a rota mais eficiente a ser seguida.

Aplicando a abordagem de força bruta à nossa topologia com 9 nós, o número total de rotas possíveis será (n-1)!, com n sendo a quantidade de nós e a subtração por um (1) significando o nó de partida que é retirado antes de gerar as permutações. Com isso, o nosso cenário contará com (9-1)! = 40320 possibilidades. Isso significa que o algoritmo de força bruta terá que calcular a distância para cada uma dessas 40320 rotas e encontrar a rota mais curta.

Essa abordagem permite uma análise completa das possibilidades e resulta na identificação da rota ótima. No entanto, devido à natureza do algoritmo de força bruta, que testa todas as combinações de rotas possíveis, o tempo de execução desse algoritmo cresce exponencialmente com o aumento do número de nós. Isso significa que, para problemas com muitos nós, o algoritmo de força bruta pode se tornar impraticável devido ao tempo excessivo necessário para encontrar a solução ótima. Portanto, é importante considerar outras abordagens e algoritmos para resolver o problema do caixeiro viajante em casos com um grande número de nós.

3.1. Complexidade do Algoritmo

Uma análise da complexidade do algoritmo de força bruta é fundamental para entender o desempenho e as limitações dessa abordagem na resolução do Problema do Caixeiro Viajante. A complexidade computacional do algoritmo está diretamente relacionada ao número de permutações a serem testadas.

Dentro do algoritmo de força bruta, a geração de todas as permutações requer um esforço computacional significativo. O cálculo de todas as distâncias entre os nós também contribui para o tempo de execução do algoritmo. Assim, a complexidade do algoritmo de força bruta pode ser expressa como O((n-1)!), ou seja, fatorial de n-1.

Essa complexidade fatorial indica que o tempo de execução do algoritmo cresce rapidamente à medida que o número de nós aumenta. Por exemplo, se tivéssemos 10 nós, o número de permutações seria (10-1)! = 362880. Para 11 nós, teríamos (11-1)! = 3.991.680 permutações, e assim por diante. Aumentos relativamente pequenos no número de nós resultam em um aumento exponencial no número de permutações e no tempo necessário para encontrar a solução ótima.

4. Conclusão

No presente artigo, é apresentado um estudo aprofundado do problema do caixeiro viajante, aplicado ao contexto da caça aos Pokémons na região da Universidade Federal do Piauí. O problema do caixeiro viajante é um desafio clássico na área de otimização combinatória, e sua aplicação nesse contexto específico oferece uma excelente maneira de otimizar a jornada dos caçadores, minimizando a distância percorrida.

Uma abordagem de solução utilizada foi a força bruta, a qual consiste em testar todas as possíveis permutações de visitação dos pontos de interesse, calculando a distância total percorrida em cada iteração. Embora seja uma solução exata, a eficiência da abordagem diminui rapidamente à medida que o número de pontos aumenta, devido ao crescimento exponencial do número de permutações possíveis.

Este estudo fornece uma visão abrangente do problema do caixeiro viajante aplicado à caça aos Pokémons na região da Universidade Federal do Piauí. Além disso, é possível estender a aplicação desse problema a outros contextos e regiões, buscando otimizar diferentes tipos de jornadas e percursos. Em suma, o problema do caixeiro viajante continua sendo um desafio relevante e em constante evolução, com amplas possibilidades de aplicação e pesquisa.

Referências

[1] Wikipedia. Problema do Caixeiro Viajante. `https://pt.wikipedia.org/wiki/Problema_do_caixeiro-viajante`. Acessado em 18 de junho de 2023.

[2] Thomas Bäck. Evolutionary algorithms in theory and practice, 1996.
[3] Localização no google maps. https://goo.gl/maps/LxCK8f1mRYJX3Jgq9. Acessado em 18 de junho de 2023.
[4] Osvaldo Carvalho. O Problema do Caixeiro Viajante. https://www.youtube.com/watch?v=CJm4mo1cUjM. Acessado em 19 de junho de 2023.