Análise e Aplicação do Algoritmo de Caminhos Mínimos de Floyd-Warshall

Marcos de Oliveira Campos
Ciência da Computação
Unieuro
Brasília, Brasil
marcos.oliveira@unieuro.edu.br

Oliver Henrique Ferreira de Jesus
Ciência da Computação
Unieuro
Brasília, Brasil
oliver.jesus@unieuro.edu.br

Eliane Freitas Borges
Ciência da Computação
Unieuro
Brasília, Brasil
eliane079695@unieuro.com.br

Matheus Nery Walkowicz
Ciência da Computação
Unieuro
Brasília, Brasil
matheus077651@unieuro.com.br

Aldo Henrique Dias Mendes
Ciência da Computação
Unieuro
Brasília, Brasil
aldo.mendes@unieuro.edu.br

***Abstract*— Este documento explora o problema do caminho mínimo em grafos, com foco no algoritmo de Floyd-Warshall. O trabalho aborda os conceitos teóricos de grafos, a análise de complexidade do algoritmo e sua aplicação prática através de uma implementação em linguagem C que simula um sistema de GPS para cálculo de rotas entre cidades. O objetivo é apresentar os conceitos, a análise do algoritmo e um exemplo de uso prático, conforme os requisitos da disciplina de Estruturas de Dados.**

I. INTRODUÇÃO

Estruturas de dados do tipo grafo são utilizadas para modelar relações entre objetos. Um grafo é composto por um conjunto de vértices (ou nós) e um conjunto de arestas (ou arcos) que conectam pares de vértices. Uma aplicação comum é a representação de redes, como mapas de cidades (vértices) interligadas por estradas (arestas).

Um problema recorrente nesse contexto é encontrar o caminho de menor custo entre dois vértices em um grafo ponderado, onde cada aresta possui um peso associado (como distância ou tempo). Este trabalho foca nos algoritmos de caminhos mínimos, especificamente o algoritmo de Floyd-Warshall, que se destaca por calcular a distância mínima entre todos os pares de vértices de um grafo. A análise abrange sua base teórica, complexidade e uma demonstração prática.

II. CONCEITOS E PROPRIEDADES DE GRAFOS

Um grafo G é formalmente definido como um par (V,E), onde V é um conjunto de vértices e E é um conjunto de arestas. Em um grafo ponderado, uma função de peso w:ErightarrowmathbbR atribui um valor a cada aresta.

Para a implementação computacional, grafos podem ser representados de diversas formas. Uma abordagem comum é a matriz de adjacência, que consiste em uma matriz M de dimensões |V|times|V|, onde a posição M[i][j] armazena o peso da aresta entre os vértices i e j. Se não houver uma aresta direta, o valor pode ser infinito (∞). Esta representação é particularmente eficiente para grafos densos e é a base para o algoritmo de Floyd-Warshall.

III. O ALGORITMO DE FLOYD-WARSHALL

O algoritmo de Floyd-Warshall é um método de programação dinâmica que resolve o problema de caminhos mínimos para todos os pares de vértices (all-pairs shortest path). Ele funciona de maneira iterativa, considerando progressivamente todos os vértices como intermediários no caminho entre outros dois.

A. Funcionamento e Análise de Complexidade

O algoritmo opera sobre uma matriz de distâncias, que é inicializada com os pesos das arestas diretas. Em seguida, ele executa três laços aninhados:

Pseudo Código:

```
para k de 0 até n - 1 faça
    para i de 0 até n - 1 faça
        para j de 0 até n - 1 faça
            se dist[i][k] + dist[k][j] < dist[i][j] então
                dist[i][j] ← dist[i][k] + dist[k][j]
            fim se
        fim para
    fim para
fim para
```

(Figura 1. Adaptado de Cormen et. al. 2009).

Devido a essa estrutura de laços triplamente aninhados, onde
n é o número de vértices, a análise de complexidade de tempo do algoritmo é $O(n^3)$. A complexidade de espaço é de $O(n^2)$ para armazenar a matriz de adjacência.

B. Análise Comparativa

Em comparação com outros algoritmos, como o de Dijkstra, o Floyd-Warshall apresenta uma complexidade de tempo maior. O algoritmo de Dijkstra, quando executado para todos os vértices usando um heap binário, teria uma complexidade de $O(V \cdot (E + V \log V))$. No entanto, o Floyd-Warshall se beneficia por sua simplicidade de implementação e por ser capaz de lidar com arestas de peso negativo (desde que não existam ciclos de peso negativo), uma limitação do algoritmo de Dijkstra.

IV. APLICAÇÕES PRÁTICAS

A capacidade do Floyd-Warshall de pré-calcular todas as rotas o torna adequado para sistemas onde as consultas de caminhos são frequentes e o grafo não muda constantemente. Alguns exemplos de uso prático incluem:

Sistemas de Navegação (GPS): Calcular a distância mais curta entre quaisquer duas cidades em um mapa.

Roteamento de Redes: Determinar o caminho mais rápido para pacotes de dados entre todos os nós de uma rede.

Bioinformática: Análise de similaridade entre sequências de proteínas.

A seguir, descrevemos uma implementação que demonstra uma dessas aplicações.

V. DESCRIÇÃO DA IMPLEMENTAÇÃO

Para o trabalho, foi desenvolvida uma aplicação em linguagem C que simula um sistema de GPS. A implementação utiliza as seguintes estruturas:

struct Location: Armazena os dados de cada vértice (cidade), como ID, nome e coordenadas.

distance_matrix[][]: Uma matriz de adjacência para armazenar as distâncias diretas entre as cidades, inicializada com ∞ para cidades não conectadas.

next_matrix[][]: Matriz auxiliar para reconstruir o caminho mínimo entre dois pontos.

O programa principal inicializa o sistema com um conjunto de cidades e estradas do Brasil. Em seguida, a função floyd_warshall(&gps) é executada uma única vez para calcular as distâncias mínimas entre todas as localizações. Após o cálculo, o usuário pode interativamente solicitar a rota entre uma cidade de origem e uma de destino, e o programa retorna a distância total e o caminho detalhado, consultando as matrizes pré-calculadas. Esta implementação serve como um exemplo prático de como o algoritmo é aplicado, demonstrando seu resultado final em um cenário de uso real.

VI. CONCLUSÃO

O estudo e a implementação do algoritmo de Floyd-Warshall permitiram aprofundar o conhecimento sobre algoritmos de caminhos mínimos em grafos. A análise de sua complexidade,
O(n3), evidencia a relação de troca entre tempo de processamento e a conveniência de ter todas as rotas mínimas pré-calculadas.

A aplicação prática desenvolvida demonstrou a viabilidade do algoritmo para sistemas que requerem múltiplas consultas de rotas em um grafo estático. O trabalho cumpre os objetivos de explorar conceitos teóricos, realizar a análise de complexidade e apresentar um exemplo de uso, integrando os subtemas propostos para a disciplina.

REFERÊNCIAS

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, 3rd ed. MIT Press, 2009.

[2] R. Sedgewick and K. Wayne, Algorithms, 4th ed. Addison-Wesley, 2011.

[3] D. E. Knuth, The Art of Computer Programming, Volume 1: Fundamental Algorithms, 3rd ed. Addison-Wesley, 1997.