



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Ciências Exatas e de Informática

André Luiz Rocha Cabral¹

Douglas Nicolas Silva Gomes²

João Paulo Dias Estevão³

Ricardo Soares Pereira da Gama Cerqueira⁴

Victor Souza Lima⁵

Yago Almeida Melo⁶

Resumo

Este relatório apresenta a implementação e análise comparativa de duas abordagens para o problema de enumeração de ciclos em grafos não direcionados simples. As abordagens analisadas são: (i) uma baseada na permutação dos vértices e (ii) outra baseada em caminhamento (DFS) no grafo. São discutidos os detalhes da implementação, a escolha da representação do grafo e os experimentos realizados para avaliar o desempenho dos algoritmos. Ademais, discute-se a aplicabilidade das implementações para grafos direcionados.

Palavras-chave: Grafos. Permutação. Caminhamento.

¹Código (Permutação)

²Código (Caminhamento)

³Código (Caminhamento)

⁴Relatório

⁵Código (Permutação)

⁶Relatório

1 INTRODUÇÃO

A enumeração de ciclos em grafos é um problema central em diversas áreas, como análise de circuitos, bioinformática e redes de comunicação. Um ciclo é definido como um caminho fechado sem repetição de vértices (exceto o primeiro e o último, que são iguais). A detecção e listagem de tais ciclos podem ser realizadas por diferentes abordagens, cada uma com suas vantagens e limitações em termos de complexidade e escalabilidade.

2 METODOLOGIA E IMPLEMENTAÇÃO

Neste trabalho, foram propostas duas abordagens para resolver o problema:

2.1 Abordagem Baseada em Permutação de Vértices

Esta abordagem consiste em gerar todas as permutações possíveis dos vértices e, para cada sequência, verificar se forma um ciclo válido. Apesar de sua simplicidade conceitual, a complexidade combinatória torna esta técnica inviável para grafos com um número moderado ou elevado de vértices. Na implementação desta abordagem, são realizadas as seguintes operações:

- Geração de todas as permutações dos vértices.
- Para cada subvetor de tamanho maior ou igual a 3, verificação se os vértices formam um ciclo, ou seja, se cada par consecutivo de vértices possui uma aresta entre si e o último está conectado ao primeiro.
- Normalização dos ciclos encontrados (através de rotação e inversão) para evitar duplicatas.

Para essa abordagem, foram utilizados para estudo e referência: (ABSTRATA, 2020) (COSTA, 2021) (FEOFILOFF, 2017) (CURSA, 2023)

2.2 Abordagem Baseada em Caminhamento (DFS)

A abordagem implementada utiliza uma busca em profundidade (DFS) modificada para detectar ciclos durante o percurso. Os pontos-chave desta implementação são:

- **Representação do Grafo:** Utilizou-se uma lista de adjacência para representar o grafo, associada a um `std::map` que mapeia cada vértice para um caractere (por exemplo, 0

para 'A', 1 para 'B', etc.). Essa escolha permite uma manipulação eficiente, especialmente em grafos esparsos.

- **Deteção de Ciclos:** Durante o DFS, ao visitar um vértice, o algoritmo marca-o como visitado e o adiciona ao caminho atual. Caso um vértice adjacente já esteja presente no caminho (e não seja o pai imediato), um ciclo é identificado.
- **Eliminação de Duplicatas:** Para evitar a contagem de ciclos iguais (mesmo que apresentados em rotações ou ordem inversa), o algoritmo compara cada ciclo recém-encontrado com os ciclos previamente armazenados.

O arquivo `main.cpp` exemplifica essa abordagem.

Para essa abordagem, foram utilizados para estudo e referência: (CONTE et al., 2016) (SCH, 2018) (FEOFILOFF, 2019) (LIU; WANG, 2006)

3 EXPERIMENTOS E RESULTADOS

Foram realizados experimentos em grafos com diferentes tamanhos e densidades, com o intuito de comparar o desempenho das duas abordagens. Os principais achados foram:

- **Eficiência da Abordagem DFS:** A técnica baseada em DFS mostrou-se significativamente mais eficiente, pois explora de maneira direcionada apenas os caminhos possíveis, reduzindo o número de verificações necessárias.
- **Limitações da Abordagem por Permutação:** A abordagem por permutação, embora simples, sofre com a explosão combinatória de possibilidades, tornando-a impraticável para grafos com muitos vértices.

Além disso, a escolha da lista de adjacência como estrutura de dados foi decisiva para a eficiência da abordagem DFS, possibilitando uma iteração rápida sobre os vizinhos de cada vértice.

4 DISCUSSÃO

Ambas as implementações apresentadas têm sucesso para grafos não direcionados simples. Contudo, para estender os algoritmos a grafos direcionados, seria necessário ajustar a lógica de verificação dos ciclos para considerar a direção das arestas. Em grafos direcionados, os critérios de conexão devem respeitar a orientação dos arcos, exigindo, possivelmente, novas técnicas de verificação e eliminação de duplicatas.

5 CONCLUSÃO

Este trabalho comparou duas abordagens para a enumeração de ciclos em grafos, concluindo que a abordagem baseada em DFS é a mais adequada para grafos de maior dimensão, dada a sua eficiência computacional. A representação por lista de adjacência se mostrou eficaz, e os experimentos realizados confirmam a superioridade da técnica DFS sobre a abordagem por permutação. Pesquisas futuras poderão explorar otimizações na detecção de ciclos duplicados e adaptações para o tratamento de grafos direcionados.

REFERÊNCIAS

ABSTRATA, Canal da Álgebra. **Aula 03 Parte II: Grupos de Permutação: Ciclos**. 2020. Disponível em: <<https://www.youtube.com/watch?v=AE5ACJs9hGM>>.

CONTE, Alessio et al. Enumerating Cyclic Orientations of a Graph. In: **26th International Workshop on Combinatorial Algorithms, IWOCA 2015**. Verona, Italy: [s.n.], 2016. v. 9538, p. 88 – 99. Disponível em: <<https://inria.hal.science/hal-01557026>>.

COSTA, André. **Grupos de Permutações - Toda permutação é composição de ciclos disjuntos**. 2021. Disponível em: <<https://www.youtube.com/watch?v=sk01TYGBZzw>>.

CURSA. **Combinatória e Teoria dos Grafos: Conectando Conceitos de Grafos e Contagem**. 2023. Disponível em: <<https://cursa.app/pt/artigo/combinatoria-e-teoria-dos-grafos-conectando-conceitos-de-grafos-e-contagem>>.

FEOFILOFF, Paulo. **Ciclos e grafos acíclicos**. 2017. Disponível em: <https://www.ime.usp.br/~pf/algoritmos/_em/_grafos/aulas/dag.html>.

FEOFILOFF, Paulo. **Ciclos e dags**. 2019. Disponível em: <https://www.ime.usp.br/~pf/algoritmos/_para/_grafos/aulas/cycles-and-dags.html>.

LIU, Hongbo; WANG, Jiaxin. A new way to enumerate cycles in graph. In: **Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services (AICT-ICIW'06)**. [S.l.: s.n.], 2006. p. 57–57.

SCH, Philipp. **Enumerating All Cycles in an Undirected Graph**. 2018. Disponível em: <<https://www.codeproject.com/Articles/1158232/Enumerating-All-Cycles-in-an-Undirected-Graph>>.