



ALUNO: Lucas Bottrel Lopes de Moura, Fernanda Ribeiro Passos Cirino

Neste trabalho buscamos elaborar algoritmos de busca de ciclos em um grafo não direcionado. Com isso, como foi solicitado na elaboração da tarefa, tentamos solucionar o problema com duas abordagens diferentes: Através de caminhamento no grafo e por permutação. Pegamos como exemplo para comparação o grafo dado:

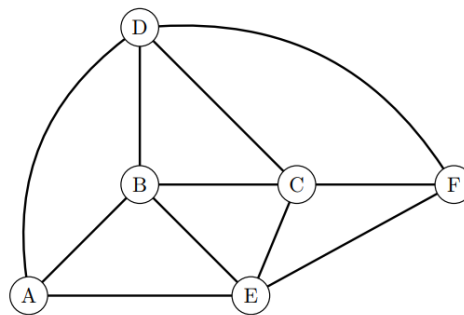


Figura 1: Exemplo de grafo com 6 vértices e 11 arestas.

1. Busca ciclos por Caminhamento

O algoritmo criado para buscar os ciclos em um grafo através do caminhamento tem como princípio realizar buscas em profundidade por cada vértice do grafo, a fim de encontrar caminhos que terminavam com o vértice de origem, e assim escrever aquele caminho. Ao todo, encontramos 38 ciclos no grafo exemplo, em um tempo médio de 9ms de execução.

2. Busca ciclos por Permutação

A ideia dessa implementação é gerar todos caminhos possíveis, dadas as arestas do grafo, e filtrar através de um algoritmo que verifica se aquele caminho é um ciclo, partindo do pressuposto de que o vértice da primeira aresta é igual ao vértice da última aresta. Realizado o filtro, teríamos os ciclos do grafo.

Infelizmente nesta implementação tivemos dificuldade em implementar o filtro corretamente, com isso temos um resultado inconclusivo numericamente. Por tanto, sabendo que a permutação de todos os vértices é um trabalho demorado, assim como filtrar e analisar cada caminho possível no grafo, concluímos que seria um algoritmo mais caro e mais demorado, certamente ultrapassando o tempo médio do algoritmo anterior (por busca em profundidade).

3. Conclusão

Apesar de não termos conseguido um resultado muito claro em relação ao algoritmo por permutação, concluimos que o algoritmo de busca em profundidade gera resultados mais rápido e com menos custo computacional, uma vez que o outro método (por permutação) deve gerar 396.990 possibilidades de caminhos no grafo e filtrar cada um desses caminhos para validar ciclos.