AED III - Trabalho Prático 1 - Demonstração de que o Ciclo Hamiltoniano é NP-Completo:
Ciclo Hamiltoniano

Elaborado por

José Argemiro dos Reis Neto (2021.1.08.011)
Otávio Augusto Marcelino Izidoro (2018.1.08.041)
Pedro Augusto Mendes (2021.1.08.041)
Rikson Pablo Gomes da Rocha (2022.2.08.007)
Victor Ribeiro Gonçalez (2021.1.08.023)

Como matéria do curso
Algoritmo e Estrutura de dados III (DCE 529)
Professor(a)
Iago Augusto de Carvalho

Curso de Ciência da Computação
Universidade Federal de Alfenas

10 de Março 2024

1 Introdução

O problema do Ciclo Hamiltoniano é um clássico exemplo de complexidade NPcompleto, o que quer dizer que ele é um dos problemas mais difíceis dentro da classe NP (classe dos problemas de decisão que podem ser resolvidos com uma máquina de Turing não-determinística em tempo polinomial). Nesse artigo será apresentado uma demonstração de que o problema citado realmente é NPcompleto e ocasionalmente ele será chamado de CICLO-HAM.

1.1 Problema

Dado um grafo G(V, E), em que V representa os vértices do grafo e E representa as arestas, queremos determinar se G contém um Ciclo Hamiltoniano (grafo em que todos os vértices são visitados uma única vez e o último vértice é adjacente ao primeiro vértice visitado). Para provar que o problema do Ciclo Hamiltoniano é NP-completo deve-se verificar que: 1. CICLO-HAM é NP 2. CICLO-HAM tambem é NP-difícil E para provar que o CICLO-HAM ´e NP-difícil será realizado uma redução de outro problema já conhecido como NP-Completo, o Caminho Hamiltoniano (que será chamado de CAM-HAM ao longo do texto), em tempo polinomial para o problema que estamos tentando provar de forma que um grafo G possui um caminho hamiltoniano se e somente se G' tiver um ciclo hamiltoniano.

2 Demonstração

2.1 O Ciclo Hamiltoniano está em NP

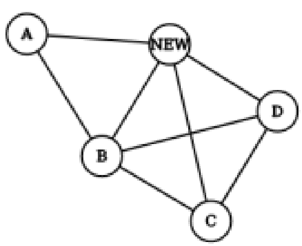


Figura 1: Primeira imagem

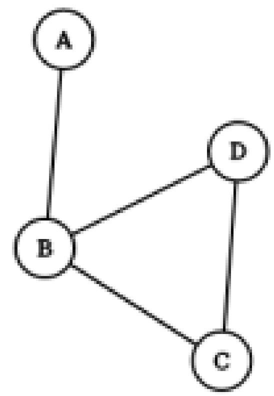


Figura 2: Segunda imagem

Primeiramente, para algum problema estar em NP, deve existir um "certificado" (uma solução) conhecido para o problema e um "verificador" que é um 1 algoritmo que realiza a verificação da solução. Caso o verificador execute em tempo polinomial a partir do certificado como entrada, pode-se dizer que esse problema está em NP. No caso do CICLO-HAM o certificado pode ser um vetor contendo os vértices em ordem e o verificador deve executar um algoritmo que atravessa esse vetor analisando se passou apenas uma vez por cada vértice, se cada par de vértices são adjacentes entre si e se o último vértice é adjacente ao primeiro (para formar um ciclo). Como esse algoritmo executa em tempo polinomial O(V + E), pode-se concluir que o CICLO-HAM NP.

2.2 O Ciclo Hamiltoniano é NP-Difícil

O segundo passo da demonstração é reduzir o Caminho Hamiltoniano ao Ciclo Hamiltoniano, ou seja, CAM-HAM p CICLO-HAM. Seja um grafo G(V, E) um grafo contendo um caminho hamiltoniano qualquer. E possível transformá-lo em um ciclo Hamiltoniano assumindo um grafo G(V, E) em que o número de vértices é dado por V = V + 1 (adiciona-se um novo vértice Vnew a G') e o número de arestas é dado por E = E + V (adiciona-se uma aresta a Vnew para cada vértice V de G). Assim, se G tem um caminho hamiltoniano, pode-se partir de qualquer vértice no caminho, percorre-lo e terminar no último vértice V final, que ao realizar a redução pode-se formar um ciclo hamiltoniano em G' conectando o V final ao Vnovo e o Vnovo ao primeiro vértice da sequência. Agora, para a direção somente se, deve-se supor que G' tem um Ciclo Hamiltoniano. Uma vez que o ciclo visita cada vértice em G', exceto V novo, pode-se remover V novo para obter um Caminho Hamiltoniano em G e portanto está provado um problema NP-completo pode ser reduzido ao ciclo hamiltoniano e por isso ele é NP-Difícil.

3 Exemplo

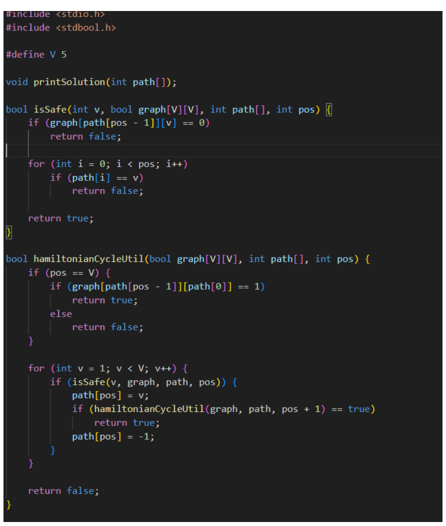
Neste codigo abaixo em na linguagem C utilizamos backtracking. É um algoritmo de força bruta e pode não ser eficiente para grafos grandes. Este programa em C define uma função `hamiltonianCycle` que tenta encontrar um ciclo hamiltoniano em um grafo representado por uma matriz de adjacências. Se encontrar um ciclo, ele imprime a solução.

O grafo é representado por uma matriz de adjacências `graph`, onde `graph[i][j]` é verdadeiro se houver uma aresta entre o vértice `i` e o vértice `j`. A função `hamiltonianCycle` inicializa um array `path` com -1 e tenta construir o ciclo hamiltoniano começando do vértice 0. A função `hamiltonianCycleUtil` é usada para resolver o problema de forma recursiva.

A função `isSafe` verifica se um vértice pode ser adicionado ao `path`. Um vértice `v` pode ser adicionado ao `path` se:

1. Existe uma aresta do último vértice adicionado ao `v`.

2. `v` não está no `path` atual.



```c
#include <stdio.h>
#include <stdbool.h>

#define V 5

void printSolution(int path[]);

bool isSafe(int v, bool graph[V][V], int path[], int pos) {
    if (graph[path[pos - 1]][v] == 0)
        return false;

    for (int i = 0; i < pos; i++)
        if (path[i] == v)
            return false;

    return true;
}

bool hamiltonianCycleUtil(bool graph[V][V], int path[], int pos) {
    if (pos == V) {
        if (graph[path[pos - 1]][path[0]] == 1)
            return true;
        else
            return false;
    }

    for (int v = 1; v < V; v++) {
        if (isSafe(v, graph, path, pos)) {
            path[pos] = v;
            if (hamiltonianCycleUtil(graph, path, pos + 1) == true)
                return true;
            path[pos] = -1;
        }
    }

    return false;
}
```

Figura 3: Primeira imagem

```c
bool hamiltonianCycle(bool graph[V][V]) {
    int path[V];
    for (int i = 0; i < V; i++)
        path[i] = -1;

    path[0] = 0;
    if (hamiltonianCycleUtil(graph, path, 1) == false) {
        printf("Solução não existe.\n");
        return false;
    }

    printSolution(path);
    return true;
}

void printSolution(int path[]) {
    printf("Ciclo Hamiltoniano encontrado: ");
    for (int i = 0; i < V; i++)
        printf("%d ", path[i]);
    printf("%d\n", path[0]);
}

int main() {
    bool graph1[V][V] = {{0, 1, 0, 1, 0},
                         {1, 0, 1, 1, 1},
                         {0, 1, 0, 0, 1},
                         {1, 1, 0, 0, 1},
                         {0, 1, 1, 1, 0}};

    hamiltonianCycle(graph1);

    return 0;
}
```

Figura 4: Segunda imagem

4 Conclusão

A partir do que foi demonstrado até agora, fica claro que o problema do Ciclo Hamiltoniano é da classe NP e também é NP-diícil, o que demonstra que tal problema é NP-Completo.

5 Referências

Os conteúdos acessados e analisados para a elaboração deste trabalho escrito foram:

PROVA DE QUE O CICLO HAMILTONIANO É NP-COMPLETO
https://acervolima.com/prova-de-que-o-ciclo-hamiltoniano-e-np-completo/
Acesso em 08 de Março de 2024.

Caminho hamiltoniano
https://pt.wikipedia.org/wiki/Caminho$_{h}amiltoniano/$
$Acessoem$08$deMarode$2024.

Backtracking
https://pt.wikipedia.org/wiki/Backtracking/
Acesso em 08 de Março de 2024.

Pesquisa Operacional II - Aula 26 - Caminhos e ciclos Euleriano e Hamiltoniano
https://www.youtube.com/watch?v=YWQQ-7U0PzU
Acesso em 08 de Março de 2024.