UNIVERSIDADE FEDERAL DA PARAÍBA

CENTRO DE INFORMÁTICA (CI)

CURSO ENGENHARIA DE COMPUTAÇÃO

EPITÁCIO PESSOA DE BRITO NETO

11506856

GIOVANNI BRUNO TRAVASSOS DE CARVALHO

11506849

PROBLEMA DE OTIMIZAÇÃO – COLORAÇÃO DE GRAFOS (GRAPH COLOURING)

Professor: Gilberto Farias

JOÃO PESSOA

2019

1. Descrição do problema

A partir de um grafo qualquer, temos que uma coloração de grafos é, basicamente, categorizar elementos pertencentes ao mesmo (vértices ou arestas). Ao atribuirmos significados em elementos de um grafo, podemos demonstrar/ilustrar informações de diversas utilidades, como traçar rotas de metro, separar continentes por cor em mapas-mundi, mostrar as rotas de entrega de um sistema de correios, entre outras aplicações.

Temos, então, duas maneiras de categorizarmos (de forma geral) o método de coloração de grafos: a coloração de vértices e coloração de arestas. Tratando da coloração de vértices, podemos descrever abordagens para realização da coloração, um deles sendo o método sequencial. Podemos classificar esta metodologia como uma heurística gulosa, e um dos pensamentos que podemos considerar é: garantir um limite superior em relação ao numero de cores, mas não garantir um uso mínimo de cores. Desta forma, podemos representar este método a partir do pseudo-código, onde Ci é uma classe de cor, i é um numero de 1 até n, xi são os vértices que, no inicio, não possuem cor. Temos, então:

$Algoritmo\ CS\ (G = (N, \Gamma))$	
1	para $i = 1$ até n faça
2	$C_i \leftarrow \emptyset$
3	fim-para
4	$k \leftarrow 1$
5	para $i = 1$ até n faça
6	enquanto $não\ atribuir x_i\ a\ C_k$ faça
7	se $\Gamma(x_i) \cap C_k = \emptyset$ então
8	$C_k \leftarrow C_k \cup \{x_i\}$
9	senão
10	$k \leftarrow k + 1$
11	fim-se
12	fim-enquanto
13	$k \leftarrow 1$
14	fim-para

Figura 1: Algoritmo de coloração de grafos de complexidade sequencial.

Se denotássemos valores para os vértices, podemos ver claramente a decisão "gulosa" a partir da linha 7, ao utilizarmos a interseção para identificar a compatibilidade com a cor. Também podemos perceber que, dependendo de como rotulamos os vértices, possuiríamos diferentes resultados no grafo final. "Se sucessivas rotulações forem realizadas, inevitavelmente a coloração mínima *(3-coloração)* será obtida, porém existem n! possíveis rotulações, o que implica num tempo de execução exponencial". Sabendo que o tempo de execução, no pior caso, assume O(exp), faremos uma classificação do problema, com finalidade de mostrarmos que a *Coloração de Grafos* é um algoritmo que pertence a NP.

2. Prova que pertence a NP

Dado o algoritmo sequencial para coloração de grafos anteriormente, devemos enxergar pela ótica de encontrar um equivalente de problema decisão para o mesmo. Por exemplo, considere o problema objetivo: dado um grafo, colorir o primeiro vértice e, em seguida, considerar, a partir do vértice atual, uma cor de menor número que não foi utilizada pelos vértices adjacentes. Um problema decisão equivalente poderia ter o seguinte questionamento: existe um número g (onde g é menor ou igual ao número de cores utilizadas para colorir os vértices) tal que $g \leq z$ (onde z seria o mesmo parâmetro k dado em sala de aula, para não confundir com o k do pseudo-código)? Demonstrando esta resposta em tempo polinomial, então, teremos classificado o problema como NP.

```
para i = 1 até n faça
    Ci ← ∅
fim-para
k ← 1
para i = 1 até n faça
    enquanto não atribuir xi a Ck faça
        se E(xi) interseção Ck = ∅ então
            Ck ← Ck união {xi}
        senão
            k ← k + 1
        fim-se
    fim-enquanto
    k ← 1
fim-para
para i = 1 até n-1 faça
    se E(1) = E(i+1) então
        continua
    senão
        g ← g + 1
    fim-se
fim-para
se g ≤ z
    SUCESSO
senão
    FRACASSO
fim-se
```

Figura 2: Solução polinomial para problema decisão.

3. Prova que Pertence a NP-Difícil

LEMMA *Let G be a cubic, 3-edge-colored graph and $V' \subseteq V(G)$ a set of vertices of G. Let $E' \subseteq E(G)$ be the set of edges of G which connect V' to the remainder of the graph. If the number of edges of color i in E' is k_i $(i = 1, 2, 3)$, then*

$$k_1 \equiv k_2 \equiv k_3 \pmod 2$$

Proof. If E_{12} is the set of edges of G which are colored with color 1 or 2, then E_{12} consists of a collection of cycles. Thus E_{12} meets E' in an even number of edges, and so $k_1 + k_2 \equiv 0 \pmod 2$ which gives $k_1 \equiv k_2 \pmod 2$. Similarly $k_2 \equiv k_3 \pmod 2$. □

3. The components used in the construction. Given an instance C of the problem 3SAT, we will show how to construct a cubic graph G which is 3-edge-colorable if and only if C is satisfiable. The graph G will be put together from pieces or "components" which carry out specific tasks. Information will be carried between components by pairs of edges. In a 3-edge-coloring of G, such a pair of edges is said to represent the value T ("true") if the edges have the same color, and to represent F ("false") if the edges have distinct colors.

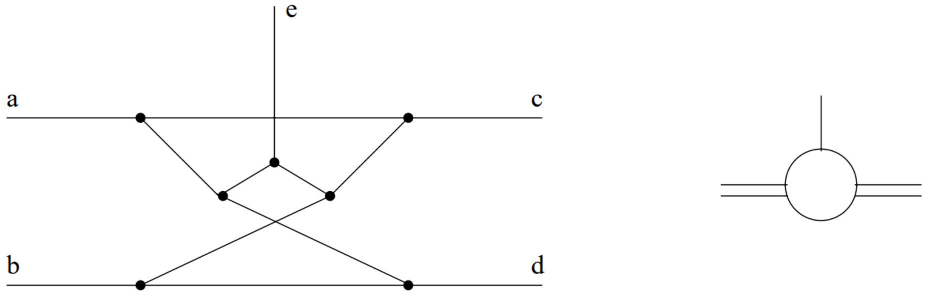

Figure 1: *The inverting component and its symbolic representation*

The inverting component is shown with its symbol in Fig. 1. It was used by Loupekine (see [4]) to construct a large family of cubic graphs with chromatic index 4. Using the parity condition above, it may be checked that if this component is 3-edge-colored, one of the pairs of connecting edges marked a, b or c, d must have equal colors and the remaining 3 edges must have distinct colors. There is no further restriction on the possible colors of the five connecting edges. Regarding the pair of edges a, b as the input and the pair c, d as the output, the component changes a representation of T to one of F and vice versa.

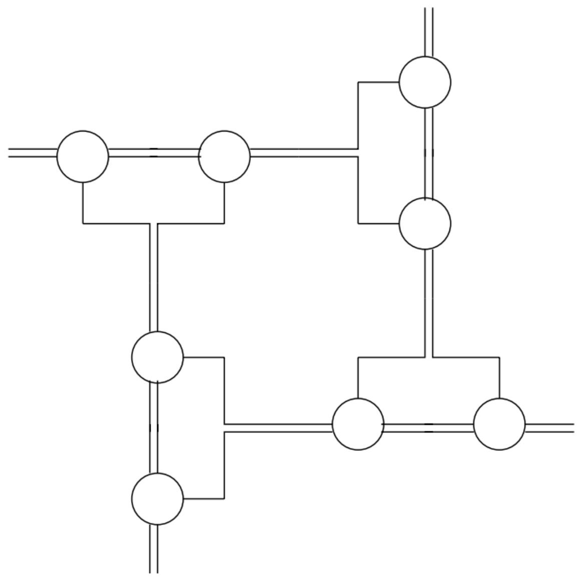

Figure 2: *The variable-setting component made from 8 inverting components and having 4 output pairs of edges. More generally, it is made from 2n inverting components and has n output pairs.*

The truth or falsity of each variable u_i will be represented by a variable-setting component such as that shown in Fig. 2. The component shown has 4 pairs of output edges, but in general the component representing u_i should have as many output pairs as there are appearances of u_i or $\overline{u}_i$ among the clauses of C. It may be checked that in any 3-edge-coloring of a variable-setting component, all the output pairs must represent the same value.

The truth of each clause c_j will be tested by a satisfaction-testing component as shown in Fig. 3. This component can be 3-edge-colored if and only if the three input pairs of edges do not all represent F. The remaining connecting edges will be discussed later.

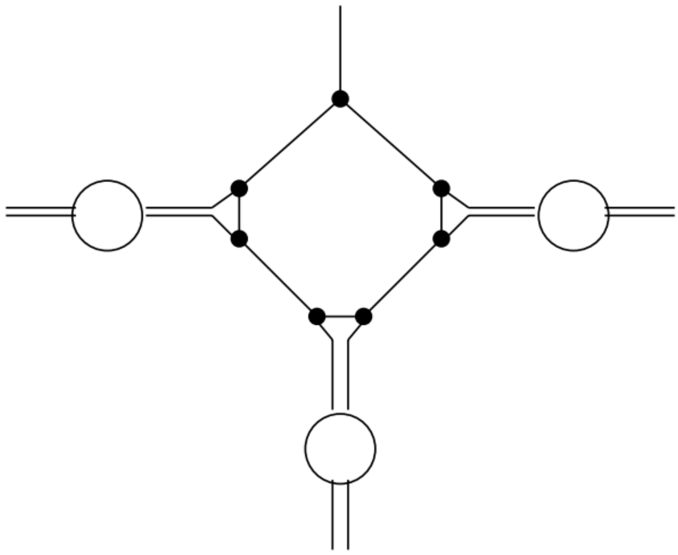

Figure 3: *The satisfaction testing component.*

THEOREM. *It is NP-complete to determine whether the chromatic index of a cubic graph is 3 or 4.*

Proof. The problem is clearly in the class NP. We exhibit a polynomial reduction from the problem 3SAT. Consider an instance C of 3SAT and construct from it a graph G as follows.

For each variable u_i take a variable-setting component U_i with one output pair of edges associated with each appearance of u_i or $\overline{u}_i$ among the clauses of C. Take also a satisfaction-testing component C_j for each clause c_j. Suppose literal $l_{j,k}$ in clause c_j is the variable u_i. Then identify the kth input pair of C_j with the associated output pair of U_i. If, on the other hand, $l_{j,k}$ is $\overline{u}_i$, then insert an inverting component between the kth input pair of C_j and the associated output pair of U_i. The resulting graph H still has some connecting edges unaccounted for. The cubic graph G is formed from two copies of H by identifying the remaining connected edges in corresponding pairs.

The graph G has a 3-edge-coloring if and only if the collection C of clauses is satisfiable, as can be verified using the properties of the components developed above. Moreover, the graph G can be produced from C using a polynomial time algorithm, so we have the result.

4. Referências

[1] A. Marie de Lima & R. Carmo, *Algoritmos Exatos para o Problema da Coloração de Grafos*, 2018.

[2] L. Mouatadid, *Introduction to Complexity Theory: 3-Colouring is NP-Complete*, 2014.

[3] R. Piacente Alves, *Coloração de grafos e aplicações*, Florianópolis-SC, 2015.

[4] I. Holyer, *The NP-Completeness of Edge-Colouring*, Reino Unido, 1981.

[5] A. Silveira Araújo Neto & M. José Negreiros Gomes, *Problema e algoritmos de coloração em grafos – exatos e heurísticos*, Salvador-BA, 2014.

[6] Notações e materiais do prof. Gilberto Farias.