UFPB

Universidade Federal da Paraíba
Centro de Informática - CI
Análise e Projetos de Algoritmos

Projeto Final
Etapa 1

Coloração de Grafos

Sabrina Alicrim Silva | 2016022764

Guilherme Moreira Rodrigues | 20160105205

Professor: Bruno Petrato Bruck

- ## Descrição do problema

Em teoria dos grafos, coloração de grafos é um caso especial de rotulagem de grafos, onde rótulos são cores. Trata-se de um problema de otimização onde procura-se diminuir a quantidade de cores necessárias para colorir um grafo. A atribuição de cores aos vértices, é sujeita a restrições, onde **dois vértices adjacentes não podem ter as mesmas cores**.

Seja G(V, U) um grafo, onde V é o conjunto de vértices e U o conjunto de arestas. Uma coloração para o grafo G é uma atribuição de cores para cada vértice de forma que vértices adjacentes tenham diferentes cores. De modo formal uma coloração consiste em função c:V(G) -> N tal que c(u) ≠ c(v), se u e v são vértices adjacentes.

- ## Prova de que pertence a NP

Para provarmos que nosso problema de coloração de grafos pertence a NP, é necessário realizar uma checagem de sua solução verificando se ela pode ser resolvida em tempo polinomial, como demanda a definição. Onde, a classe **NP(tempo polinomial não determinístico)** contém os problemas de decisão que são tidos como verificáveis em tempo polinomial.

Em outras palavras devemos responder se, dado um valor **K** pertencente, ao conjunto de cores disponíveis **N**, com essas **K** cores podemos colorir o grafo sem que seja quebrada a restrição de que vértices adjacentes não devem possuir a mesma cor.

Dado um grafo **G(V, U)** e um certificado **C**, teremos um verificador que irá verificar se a coloração **C**, segue todas as restrições e é adequada ao grafo **G**, analisando se as cores dos nós aos que cada aresta se conectam são diferentes. Para toda a aresta em **U**, levando em consideração que os vértices já estão coloridos.

Certificado: Cada nó possui uma cor, definida em um intervalo **K** de numeração para representar as cores.

Certificador: Percorrendo o conjunto de arestas **U** em **G(V, U)**

Se os vértices conectados em **U** possuem a mesma cor, a função retorna *false*, caso contrário retornará *true*.

Abaixo uma ilustração do pseudocódigo:

```
function verGColor(V, E)
        for k in E
                if k[1].color == k[2].color
                        return false;
        end
        return true;
end
```

Como podemos ver, sendo a solução validada, é feita em tempo polinomial, visto que precisou-se percorrer as arestas do grafo, para verificar as cores dos nós aos quais elas se conectam.

- ## Prova de que pertence a NP difícil

Podemos demonstrar que este problema é NP- Difícil, se consideramos que, para um problema de decisão satisfazer as condições do problema de Coloração de Grafos, sendo assim, podendo ter seus vértices coloridos por um número k de cores, e em seguida reduzi-lo a um problema anteriormente conhecido como NP- Difícil.

Desta maneira adotaremos k = 3, e reduzimos o problema para o problema SAT, conhecidamente NP-Difícil.

Reduzimos por tomar cada vértice e integralizar um array de k valores booleanos, para então definir qual cor tomará este vértice.

Em seguidas, poderemos realizar a checagem criando uma fórmula que traduz a satisfação do gráfico as condições do problema.

Assim criamos a formula que somente é verdadeira se, corresponder a uma combinação de cores válida.

Considerando k=3, ou seja, três cores diferentes para cada vértice, resultaria a seguinte formulação:

- Dado o vértice *u* fazemos, se:

$$(X1 \mid X2 \mid X3) \mathbin{\&} (\overline{X1} \mid \overline{X2} \mid X3) \mathbin{\&} (\overline{X1} \mid X3 \mid X2) \ = \ true$$

- Também será verificado se, para todos os vértices no conjunto ***U*** conectados ao vértice *u* possuem cores distintas da cor de *u*.

Desta maneira podemos concluir, se dado qualquer grafo G, é possivel colori-lo ou não com k cores distintas definidas, ou para o caso de otimização, qual seria o número k mínimo de cores para colorir o grafo G.
Como o problema SAT é considerado NP-Difícil, e reduzimos o nosso problema utilizando o SAT como base, concluiremos que o nosso problema também é NP-Difícil.

- ## **Referências:**

- http://conteudo.icmc.usp.br/pessoas/andretta/ensino/aulas/sme0216-5826-2-15/grupo5.pdf
- http://www.decom.ufop.br/menotti/paa111/files/PCC104-111-ars-11.1-MarceloFerreiraRego.pdf
- https://www.ic.unicamp.br/~atilio/slidesWtisc.pdf