

# Relatório de Resumo das aulas Aula 02

Diego Amaro Ferraz da Costa

Diego Athayde Monteiro

## **Resumo**

Neste trabalho, descreveremos as aulas ministradas pela professora Celina de Figueiredo, para a pós-graduação de Engenharia da Computação, do PESC/COPPE, baseada no material elaborado para um curso apresentado no XVII Congresso da Sociedade Brasileira de Computação em Brasília em agosto de 1997. O curso ministrado é baseado na introdução à coloração em grafos para estudando da área de ciência da computação.

# 1 Aula02

## 1.1 Introdução

O estudo da coloração de grafos se iniciou no ano de 1852, Francis Guthrie, fez o questionamento sobre o "problema das quatro cores". Esse problema consiste em avaliar se qualquer grafo planar poderia ser colorido com quatro cores. Essa questão só foi respondida por Appel e Haken [1], usando computador, após mais de 100 anos. O problema das quatro cores está relacionado a coloração dos vértices de um grafo, sendo mais preciso, corresponde à coloração mínima dos vértices de um grafo.

O problema da coloração de vértices de um grafo ou determinação do número cromático de um grafo foi definido inicialmente por Karp [2] em 1972 como um dos 21 problemas NP-completos de Karp, como são conhecidos até hoje em dia. Em sua versão mais conhecida e amplamente estudada, o problema da coloração tem como objetivo atribuir cores ou rótulos aos vértices de um grafo de tal forma que dois vértices adjacentes não possuam a mesma cor ou rótulo. Porém, este problema possui mais variantes, como a coloração de arestas ou determinação do índice cromático de um grafo, onde, de maneira semelhante a versão original, queremos atribuir cores às arestas de tal forma que arestas incidentes a um mesmo vértice (adjacentes) não possuam a mesma cor. E como uma segunda variante, existe também o problema da coloração total, onde, desejamos colorir ou atribuir rótulos simultaneamente aos vértices arestas de um dado grafo preservando todas as adjacências.

Sabemos que tanto o problema de coloração de vértices, quanto o de coloração de arestas, de um grafo são problemas NP-difíceis. Entretanto, o problema de coloração pode ser aplicado para diversas classes de grafos, e para algumas delas, temos algoritmos eficientes, leia-se polinomiais, e entre eles podemos citar os grafos bipartidos e grafos cordais, para essa última classe, ainda não se sabe se para coloração de arestas, existe um algoritmo eficiente.

O documento em questão tem por objetivo responder as 3 questões feitas em sala de aula pela professora Celina. As questões são:

- Demonstrar que, para qualquer inteiro positivo  $k$ , é possível construir um grafo sem triângulos e com número cromático  $k$ . Referência da Família dos Grafos de Mycielski;
- Mostrar um exemplo de um grafo onde o algoritmo guloso seja muito ruim, o algoritmo retorna um número de cores muito maior que o número cromático;
- Mostrar um exemplo de um grafo de comparabilidade, e um exemplo de um grafo de não comparabilidade.

## 1.2 Coloração de Vértices

Mais especificamente, uma coloração de vértices ou uma  $k$ -coloração dos vértices de um grafo  $G$  é uma associação de cores ou atribuição de rótulos aos vértices de  $G$ , utilizando  $k$ -cores. Esta  $k$ -coloração é chamada uma coloração própria, quando, de fato respeitamos a condição acima descrita: dois vértices adjacentes não podem ter a mesma cor. Nosso principal objetivo é obter o número mínimo de cores que precisamos para colorir o grafo, este chamado de número cromático e denotado por  $\chi$ . Um limite inferior natural para o problema da coloração de vértices é a clique máxima do grafo em questão, uma clique consiste em um conjunto de vértices que induzem um subgrafo completo.

No caso dos grafos bipartidos, temos este limite como uma igualdade, pois a clique máxima que pode existir em grafo bipartido é uma clique de tamanho 2, visto que não existem ciclos ímpares nesta classe de grafos.

Baseado neste limite, podemos pensar, que um grafo que possua um número cromático muito alto necessariamente tem uma clique máxima de tamanho expressivo, porém, para qualquer inteiro positivo  $k$  é possível construir um grafo com número cromático  $k$  livre de triângulos. Esta construção é chamada construção de Mycielski e pode ser descrita da seguinte forma: inicia-se com um grafo trivial  $G$  e a cada iteração são adicionados  $n + 1$  vértices a  $G$ . Cada um destes  $n$  vértices  $v_i$  é associado a um vértice  $u_i$  do grafo da iteração anterior de forma que uma aresta  $(u_1, u_2)$  origina duas novas arestas,  $(u_1, v_2)$  e  $(u_2, v_1)$ . Quanto ao  $(n + 1)$ -ésimo vértice  $w$ , este é conectado a todos os  $n$  vértices  $v_i$  que estão sendo adicionados no momento. A Figura 1 ilustra este procedimento de construção.

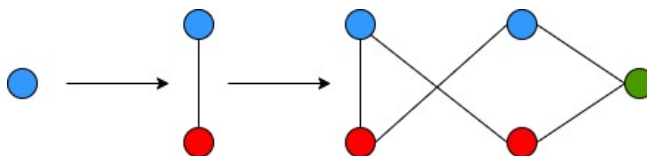


Figura 1: Construção de Mycielski. O processo aumenta em uma unidade o número cromático do grafo corrente a cada iteração, sendo assim possível a construção de um grafo livre de triângulos com qualquer número cromático.

## 1.3 Algoritmo Guloso

Apesar da dificuldade atrelada ao problema da coloração de vértices, um algoritmo guloso relativamente simples pode ser definido. O algoritmo consiste em, dada uma ordenação prévia dos vértices do grafo, atribuir uma cor ou rótulo ao vértice inicial da sequência e a cada vértice que segue na sequência, atribui-se

a cor de menor índice não utilizada ainda na sequência. É fácil notar que este algoritmo gera uma coloração válida e que também é um limite superior para  $\chi$ .

Entretanto, um detalhe bastante importante deste algoritmo é como feita esta ordenação inicial citada acima, esta ordenação. O algoritmo é sensível a escolha desta ordenação, uma que é bastante utilizada é a ordenação por ordem não decrescente dos graus dos vértices do grafo. Nem sempre esta ordenação fornece o número cromático do grafo em questão, na Figura 2 mostramos um exemplo de um grafo que estabelecidas duas ordenações, ao ser dado como entrada para o algoritmo guloso retorna um valor diferente de  $\chi$ .

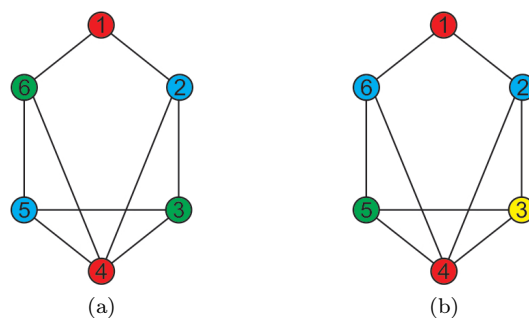


Figura 2: (a) Nesse grafo conseguimos identificar facilmente que podemos efetuar uma coloração com apenas 3 cores. Nesse caso consideramos que o algoritmo guloso utilizou a ordem de colocação 4, 5, 3, 6, 2, 1, chegando a coloração ótima. (b) Considerando o exemplo anterior, vimos uma coloração ótima, porém ainda com a mesma lógica implementada no nosso algoritmo guloso, ele pode executar a coloração em uma nova ordem, que seria 4, 6, 2, 5, 3, 1, efetuando a colocação com 4 cores, o que é diferente da coloração ótima.

O que podemos ver no exemplo acima é que o algoritmo guloso nem sempre efetuará a coloração ótima para o grafo, com isso não se tornando eficiente para o problema citado. Na figura 2a o algoritmo temos uma coloração ótima, utilizando 3 cores, na figura 2b o algoritmo não chega a coloração ótima, utilizando 4 cores para a coloração do gráfico.

Com um tipo de ordenação especial dos vértices de um grafo  $G$ , podemos obter um limite superior melhor para  $\chi(G)$ . Seja  $\mathcal{H}$  o conjunto de todos os subgrafos induzidos de  $G$ , chamamos de degenerescência de  $G$ , o maior dentre os menores graus (denotado por  $\delta(G)$ ) de cada subgrafo induzido de  $G$ . A degenerescência de  $G$  nos fornece um novo limite superior dado pela seguinte desigualdade:  $\chi \leq \text{degen}(G) + 1$

Um obstáculo para a execução do algoritmo guloso são as obstruções. Obstruções são ordens dos vértices que precisamos evitar na hora da aplicação do

algoritmo guloso. Consistem em ordens específicas de vértices que, ao submetidas ao algoritmo, retornam um valor maior que o número cromático da sequência. Quando um grafo não contém obstruções, ele admite uma ordem perfeita, a recíproca também é verdadeira. Uma ordem é perfeita, quando conseguimos uma coloração ótima através do algoritmo guloso para cada subgrafo induzido do grafo em questão, estes grafos que admitem tal ordem são chamados perfeitamente ordenáveis.

Podemos aplicar estas definições na classe dos grafos cordais. É possível definir uma ordem perfeita com base nos vértices **simpliciais** do grafo cordal. Um vértice simplicial é um vértice cuja vizinhança induz um subgrafo completo. Conseguimos então construir uma ordem perfeita de tal forma que cada vértice na sequência é simplicial em um subgrafo induzido do grafo cordal em questão. A figura abaixo representa esta sequência:

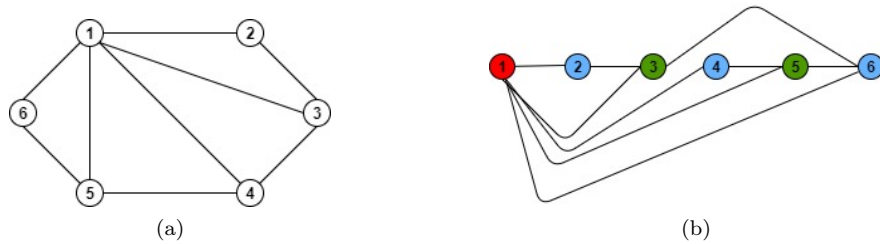


Figura 3: (a) Grafo cordal  $G$ . (b) Ordem perfeita composta pelos seus vértices simpliciais obtendo assim uma coloração ótima através do algoritmo guloso. Os vértices 1, 5 e 6 são simpliciais em  $G[1, 5, 6]$ , os vértices 2 e 3 são simpliciais em  $G[1, 2, 3]$ , e por fim, o vértice 4 é simplicial em  $G[1, 3, 4]$ .

Podemos aplicar novamente o conceito de grafos perfeitamente ordenáveis aos grafos de comparabilidade. Um grafo é de comparabilidade se admite uma ordenação transitiva em suas arestas. Esta ordenação consiste em: se existe arco  $(u, v)$  e  $(v, w)$ , então deve existir o arco  $(u, w)$ .

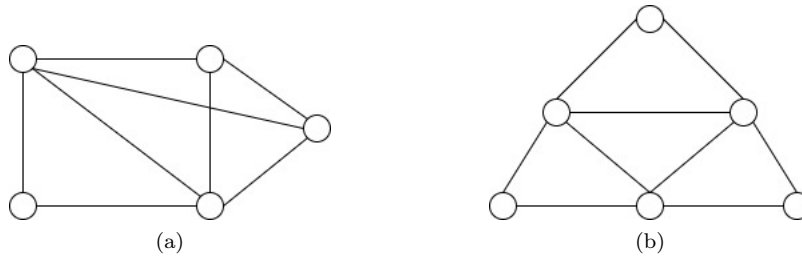


Figura 4: (a) Grafo de comparabilidade  $G$ . (b) O grafo de Hajós é um exemplo de grafo que não é de comparabilidade. Curiosamente, a remoção de qualquer vértice do mesmo o torna um grafo de comparabilidade, admitindo assim uma orientação transitiva de suas arestas.

Uma ordem perfeita pode ser elaborada tendo como base as fontes da orientação transitiva de um grafo de comparabilidade. De forma que a cada vértice na sequência é a fonte com o maior grau de saída do grafo (a cada vértice posto na sequência, o eliminamos do grafo). A Figura 5 ilustra um exemplo.

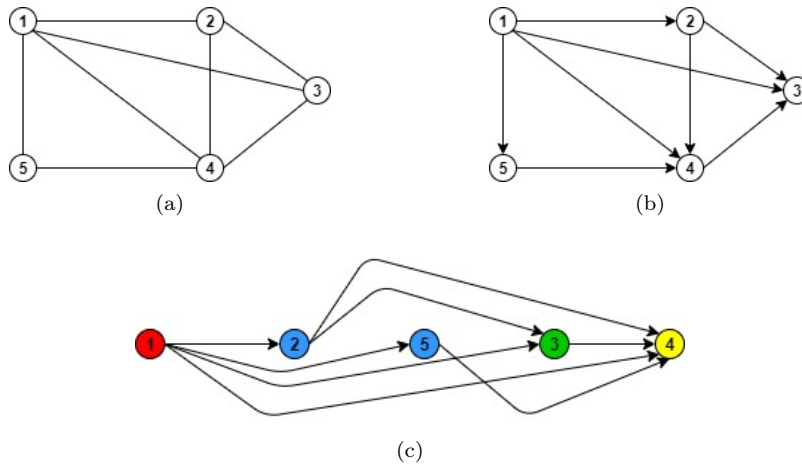


Figura 5: (a) Grafo de comparabilidade  $G$ . (b) Orientação transitiva de  $G$ . (c) Ordem das fontes que o algoritmo guloso utiliza para obter uma coloração ótima de  $G$ .

## Referências

- [1] Kenneth Appel and Wolfgang Haken. Every planar map is four colorable. *Bulletin of The American Mathematical Society - BULL AMER MATH SOC*, 82, 09 1976.
- [2] Richard Karp. Reducibility among combinatorial problems. volume 40, pages 85–103, 01 1972.