

# Utilização da técnica de coloração de grafos para escalonamento de horários dos professores da UFRR

Rafael Sá Menezes<sup>1</sup>, Rodrigo dos Santos Tavares<sup>1</sup>, Herbert Oliveira Rocha<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação  
Universidade Federal de Roraima (UFRR) – Boa Vista, RR – Brasil

{rafael.sa, rodrigo.tavares, herbert.rocha}@ufrr.br

**Abstract.** *This paper describes the use of the graph coloring technique for assigning a time schedule for professors in the Department of Computer Science in the Federal University of Roraima (UFRR). Our approach consisted in expressing subjects as vertices and time slots as colors. To represent time constraints, we used adjacencies between vertices and a list of prohibited colors in each vertex. A first-fit algorithm was used for filling the graph's colors, aiming to maximize the number of colors. In the end, we achieved great results by generating a time schedule for a real situation in polynomial time, with the complexity of  $O(|V| \times |R_T|^2)$ .*

**Resumo.** *Este artigo descreve a utilização da técnica de coloração de grafos para atribuir horários dos professores do Departamento de Ciência da Computação da UFRR. A abordagem utilizada consistiu em representar as disciplinas como vértices de um grafo e os horários disponíveis como cores. Para as restrições, utilizou-se adjacências e lista de cores nos vértices. Para preenchimento das cores, foi utilizado um algoritmo de first-fit, tentando manter sempre a maior quantidade de cores possíveis. Ao final, foram obtidos resultados satisfatórios, gerando um horário de uma situação real (o semestre 2016.1) em um tempo polinomial, com complexidade  $O(|V| \times |R_T|^2)$ .*

## 1. Introdução

No Departamento de Ciência de Computação (DCC) da Universidade Federal de Roraima (UFRR), é sempre uma tarefa complexa definir os horários das disciplinas, pois vários professores possuem certas restrições de horários – por capacitação e projetos de extensão, por exemplo. Além disso, as disciplinas ofertadas são voltadas a certas turmas de alunos e não podem ocorrer simultaneamente. Este trabalho propõe uma forma de facilitar o trabalho do DCC na hora de gerar esses horários.

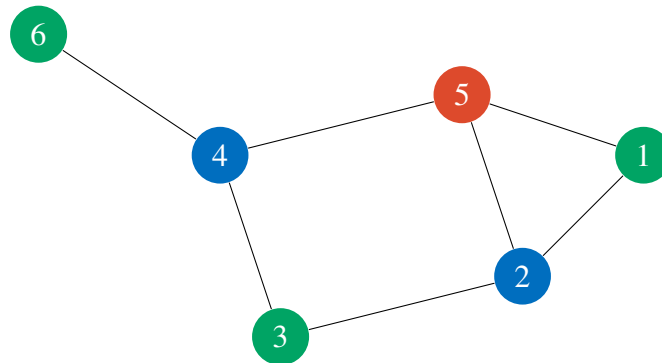
O problema de escalonamento é um problema clássico da computação, conhecida-mente NP-Completo [Cormen et al. 2001], o que tornaria muito demorado obter o horário para uma situação real. Por isso, foi utilizado um método aproximado, pois não se faz necessário obter o melhor horário possível, apenas algum que satisfaça todas as restrições dadas.

Este trabalho propõe satisfazer a seguinte pergunta: *Como gerar uma tabela de horários para todas as disciplinas do curso de Ciência da Computação da UFRR levando em conta as restrições de horários dos professores e das turmas em tempo hábil?*

## 2. Referencial Teórico

### 2.1. Coloração de grafos

A coloração de grafos é um modelo utilizado quando queremos impor certas restrições sobre os vértices de um certo grafo. No caso, os vértices adjacentes não podem ter a mesma cor. A figura 1 apresenta um exemplo de grafo colorido. [Kubale 2004]



**Figura 1. Exemplo de grafo de restrições**

## 3. Modelagem do problema

A modelagem do problema consiste em um grafo colorido onde os vértices são as disciplinas e suas cores são os horários. Pelo fato de o DCC apenas ofertar disciplinas de 60 horas no turno diurno, a semana foi separada com as seguintes cores:

- Cor A: Representa o horário 08:00 às 10:00 das segundas e quartas
- Cor B: Representa o horário 10:00 às 12:00 das segundas e quartas
- Cor C: Representa o horário 14:00 às 16:00 das segundas e quartas
- Cor D: Representa o horário 16:00 às 18:00 das segundas e quartas
- Cor E: Representa o horário 08:00 às 10:00 das terças e quintas
- Cor F: Representa o horário 10:00 às 12:00 das terças e quintas
- Cor G: Representa o horário 14:00 às 16:00 das terças e quintas
- Cor H: Representa o horário 16:00 às 18:00 das terças e quintas
- Cor I: Representa o horário 08:00 às 12:00 das sextas
- Cor J: Representa o horário 14:00 às 18:00 das sextas

Para representar as restrições da disciplina, utilizamos os vértices adjacentes para que garantíssemos que o professor não desse várias disciplinas no mesmo horário e que disciplinas voltadas para a mesma turma não fossem dadas no mesmo horário. Já as restrições do professor é representada através de uma lista auxiliar, na qual guardamos todas as cores com as quais o vértice não pode ser pintado.

O programa é dividido em duas partes: a primeira é uma interface para os usuários poderem estabelecer as disciplinas e restrições, gerando assim o grafo a ser executado posteriormente no algoritmo de coloração.

	Segunda	Terça	Quarta	Quinta	Sexta
8:00 - 10:00	DCC105 (George) DCC802 (Dion) DCC403 (Lobo)	DCC206 (Herbert) DCC602 (Balico)	DCC105 (George) DCC802 (Dion) DCC403 (Lobo)	DCC206 (Herbert) DCC602 (Balico)	DCC405 (Thais)
10:00 - 12:00	DCC205.B (Dion) DCC407 (Lobo)	DCC606 (Herbert)	DCC205.B (Dion) DCC407 (Lobo)	DCC606 (Herbert)	DCC405 (Thais)
14:00 - 16:00	DCC603 (Dion) DCC205.A (Dwan)	DCC204 (Delfa) DCC605 (Miguel)	DCC603 (Dion) DCC205.A (Dwan)	DCC204 (Delfa) DCC605 (Miguel)	
16:00 - 18:00	DCC402 (Delfa) DCC607 (Miguel)		DCC402 (Delfa) DCC607 (Miguel)		

**Tabela 1. Exemplo de horário preenchido**

### 3.1. Geração do grafo

Para geração de grafo, utilizamos uma página *web* na qual o usuário pode definir professores, disciplinas e suas restrições, com essas informações é gerado um arquivo CSV, que representa o grafo juntamente com a lista de restrições. Esse arquivo é passado para o programa principal, no qual ele é colorido e exportado via CSV, que finalmente é representado pela interface *web*.

### 3.2. Algoritmo de coloração

No algoritmo 1 é descrito o pseudocódigo da função de coloração do grafo. Analisando a função, é trivial observar que a complexidade obtida é de  $O(|V| \times |R_T|^2)$ .

---

#### Algoritmo 1 Coloração do grafo

---

```

1: função COLORCAO( $V, R_T, k$ )  $\triangleright O(|V| \times |R_T|^2)$ 
2:   Vertices  $\leftarrow$  ordenar( $V$ )  $\triangleright O(n \log n)$ 
3:   color  $\leftarrow 1$ 
4:   para todo  $v \in$  Vertices faça  $\triangleright O(|\text{Vertices}|)$ 
5:     enquanto  $\neg$ esta_pintado( $v$ ) faça  $\triangleright O(|R_T|)$ 
6:       se verifica_cor( $v, c$ ) então  $\triangleright O(|R_T|)$ 
7:          $v.\text{color} \leftarrow c$ 
8:       fim se
9:        $c \leftarrow c + 1 \bmod k$ 
10:    fim enquanto
11:  fim para
12: fim função

```

---

### 3.3. Resultados obtidos

Testamos o aplicativo com um horário parecido com o do semestre de 2016.1 e a tabela 1 mostra o resultado obtido pelo nosso programa. A execução do programa ocorreu de forma correta e os resultados não desrespeitaram nenhuma das restrições impostas anteriormente.

#### **4. Conclusão**

A abordagem utilizada para modelagem do problema dos horários no DCC – utilizando-se de coloração de grafos junto a uma lista de restrições – se mostrou ser o suficiente para atender às necessidades do problema proposto, já que em um problema real o algoritmo solucionou o problema de maneira rápida e eficiente. Uma vez que obtivemos uma complexidade relativamente baixa para a característica do problema, pudemos concluir que atingimos uma solução viável para o problema do departamento.

#### **Referências**

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., e Stein, C. (2001). *Introduction to algorithms*. McGraw-Hill Science/Engineering/Math, 2 edition.
- Kubale, M. (2004). *Graph Colorings*. Contemporary Mathematics 352. Amer Mathematical Society.