

## Coloração de Grafos (Graph Coloring)

Irving L. Santos Riça<sup>1</sup>, Marinaldo da C. Barroso Filho<sup>2</sup>, Maria V. Teixeira e  
Silva<sup>3</sup>, Rafaela M. M. da Cunha<sup>4</sup>, Carolina Y. V. Watanabe<sup>5</sup>

<sup>1</sup> Discente do curso de Graduação em Ciência da Computação, Universidade Federal de Rondônia,

Porto Velho, Brasil ([irvinglemuelricask@gmail.com](mailto:irvinglemuelricask@gmail.com))

<sup>2</sup> Discente do curso de Graduação em Ciência da Computação, Universidade Federal de Rondônia,

Porto Velho, Brasil ([barroso.marinaldo06@gmail.com](mailto:barroso.marinaldo06@gmail.com))

<sup>3</sup> Discente do curso de Graduação Computação, Universidade Federal de Rondônia, Porto

Velho, Brasil([mvitoriateixeira.s@gmail.com](mailto:mvitoriateixeira.s@gmail.com))

<sup>4</sup> Discente do curso de Graduação em Ciência da Computação, Universidade Federal de Rondônia, Porto

Velho, Brasil ([mellorafa442@gmail.com](mailto:mellorafa442@gmail.com)).

<sup>5</sup> Docente curso de Graduação em Ciência da Computação, Universidade Federal de Rondônia,

Porto Velho, Brasil ([carolina@unir.br](mailto:carolina@unir.br))

<sup>1</sup> Universidade Federal de Rondônia (UNIR) Porto Velho, RO – Brasil Rodovia BR-364, s/n  
km 9,5, Porto Velho - RO, 76801-059  
Departamento de Ciência da Computação

### Resumo

A coloração de grafos é um problema clássico da ciência da computação com aplicações em diversas áreas, como a alocação de canais de rádio. Este artigo apresenta o desenvolvimento e a avaliação de um sistema computacional baseado no algoritmo DSATUR (*Degree of Saturation*), que visa atribuir frequências a transmissores de forma a evitar interferências. A abordagem utiliza grafos para modelar os conflitos entre transmissores próximos e aplica coloração dos vértices para garantir a exclusividade das frequências. Através de testes com instâncias padronizadas da base DIMACS, os resultados indicaram eficiência na execução e uso otimizado de cores, demonstrando a aplicabilidade do algoritmo em cenários reais.

### Abstract

Graph coloring is a classical problem in computer science with applications in various fields, such as radio channel allocation. This paper presents the development and evaluation of a computational system based on the DSATUR (*Degree of Saturation*) algorithm, aiming to assign frequencies to transmitters while avoiding interference. The approach models conflicts between nearby transmitters using graphs and applies vertex coloring to ensure frequency exclusivity. Through tests with standardized instances from the DIMACS dataset, the results indicated execution efficiency and optimized color usage, demonstrating the algorithm's applicability to real-world scenarios.

# 1 INTRODUÇÃO

De acordo com os estudos sobre a Teoria dos Grafos, Leonhard Euler propôs, em 1736, um problema que marca o surgimento desse campo de estudo. Na época, discutia-se, nas ruas da cidade de Königsberg, a possibilidade de atravessar todas as pontes do local sem repetir nenhuma delas. O diagrama elaborado por Euler, com o objetivo de representar o mapa da cidade, constituiu aquilo que hoje é denominado grafo. A partir dessa abordagem inicial, diversas pesquisas e ramificações surgiram, contribuindo significativamente para o desenvolvimento científico e matemático dos conceitos relacionados aos grafos, com aplicações relevantes nas ciências exatas e biológicas (NOGUEIRA JÚNIOR, 2017).

Entre elas, destaca-se o estudo dos **grafos coloridos**, que envolve a atribuição de cores a vértices de um grafo de acordo com determinadas restrições, como a proibição de cores repetidas entre elementos adjacentes. Esses conceitos atualmente se aplicam em áreas importantes, como programação de horários, alocação de recursos e resolução de problemas de conflito.

## 1.1 Objetivo

Este artigo tem como objetivo desenvolver um sistema eficiente para a atribuição de frequências de rádio (canais) a transmissores distribuídos geograficamente, de modo a evitar interferências entre transmissores próximos. A abordagem adotada modela os transmissores como vértices de um grafo, enquanto as arestas representam possíveis conflitos de interferência entre eles. O problema consiste em colorir o grafo, atribuindo frequências (cores) aos vértices, de forma que vértices adjacentes não compartilhem a mesma frequência, garantindo assim a coexistência harmônica dos sinais e a otimização do uso do espectro disponível. Para a solução computacional, utilizou-se o algoritmo DSATUR (*Degree of Saturation*), que se mostra eficiente na resolução desse tipo de problemática.

## 2 METODOLOGIA

Essa pesquisa desenvolveu uma abordagem quantitativa baseada na análise do algoritmo denominado DSATUR (*Degree of Saturation*). Essa abordagem é mantida como gulosa, pois considera o grau de saturação dos vértices, ou seja, a quantidade de cores distintas atribuídas aos seus vértices adjacentes. De acordo com **AIRES** (2015), o processo tem início com todos os vértices não coloridos, sendo selecionado aquele com maior grau de saturação, em ordem decrescente, para receber a primeira cor. Em situações de empate, opta-se pelo vértice de maior grau; caso o empate persista, aplica-se um critério arbitrário de desempate, como a escolha do vértice de menor índice. O vértice selecionado é então colorido com a menor cor disponível que não gere conflito com seus adjacentes. Esse procedimento é repetido até que todos os vértices estejam devidamente coloridos.

### 2.1 Aplicação da Metodologia

Para aplicar o algoritmo DSATUR (*Degree of Saturation*), foi desenvolvido um sistema computacional em linguagem C (compilado com o compilador gcc ubuntu 11.4.0), estruturado com base na representação de grafos por listas de adjacência. Inicialmente, foi implementado um *parser* para leitura de arquivos no formato .col, padrão DIMACS, os quais contêm descrições formais de grafos utilizados em benchmarks acadêmicos.

Após o carregamento do grafo, a coloração foi realizada com base na metodologia DSATUR (*Degree of Saturation*), que seleciona os vértices a serem coloridos com base no grau de saturação, ou seja, na quantidade de cores distintas já atribuídas aos seus vizinhos. Conforme os testes realizados, o algoritmo apresentou desempenho consistente, alcançando soluções próximas ao limite inferior de cores, o que indica boa eficiência heurística.

A aplicação foi testada com diferentes instâncias da base DIMACS, como o arquivo inithx.i.1.col, a fim de verificar a eficiência e a escalabilidade da solução. Ao final da execução, o programa exibe no terminal a coloração final dos vértices, o número total de cores utilizadas e o tempo de processamento necessário para a resolução do problema.

Algoritmo DSATUR(Grafo g)

    Inicializar vetor cor[g.tam] com -1

    Para cada vértice j:

        q\_infos[j].indice  $\leftarrow$  j

        q\_infos[j].degree  $\leftarrow$  grau do vértice j

        q\_infos[j].sat  $\leftarrow$  0

    Ordenar q\_infos por maior saturação (sat), em caso de empate, maior grau

Enquanto existir vértice não colorido:

    vertAtual  $\leftarrow$  primeiro de q\_infos

    Remover vertAtual da fila

    u  $\leftarrow$  vertAtual.indice

Para todo i em 0 até g.tam:

    ehColorido[i]  $\leftarrow$  falso

Para cada vizinho de u:

    Se já tiver cor:

        ehColorido[cor do vizinho]  $\leftarrow$  verdadeiro

menorCor  $\leftarrow$  0

Enquanto ehColorido[menorCor] for verdadeiro:

    menorCor  $\leftarrow$  menorCor + 1

cor[u]  $\leftarrow$  menorCor

Para cada vizinho 'viz' de u ainda não colorido:

    coresUsadas  $\leftarrow$  vetor de zeros

    cont  $\leftarrow$  0

Para cada vizinho 'v' de viz:

    Se v tiver cor e coresUsadas[cor[v]] == 0:

        coresUsadas[cor[v]]  $\leftarrow$  1

        cont  $\leftarrow$  cont + 1

q\_infos[viz].sat  $\leftarrow$  cont

Reordenar q\_infos por maior saturação (sat), e grau em caso de empate

Para i de 0 até g.tam:

    Exibir: "Vértice i  $\rightarrow$  Cor cor[i]"

totalCores  $\leftarrow$  contar número de cores distintas em cor

Exibir: "Número total de cores usadas: totalCores"

FimAlgoritmo

## 2.2 Coleta de Dados

Para a análise dos dados, foi utilizada uma base contendo **instâncias de coloração de grafos** disponíveis no formato padrão **DIMACS**, do *Center for Discrete Mathematics and Theoretical Computer Science*. Essas instâncias são amplamente utilizadas em testes de algoritmos de coloração de grafos, e sua padronização facilita a leitura, interpretação e aplicação em ambientes computacionais. A base de dados encontra-se disponível: <https://mat.tepper.cmu.edu/COLOR/instances.html>.

## 3 AVALIAÇÃO EXPERIMENTAL

A avaliação experimental foi conduzida a partir da execução do algoritmo DSATUR (*Degree of Saturation*) sobre diferentes instâncias da base DIMACS, conhecida por reunir grafos utilizados como benchmark em estudos de coloração. Os testes foram realizados em diferentes arquivos .col, com grafos caracterizados como grafos leves/médios.

A implementação foi feita em linguagem C, considerando a leitura de arquivos no formato .col, a construção de grafos por listas de adjacência e a aplicação direta do algoritmo DSATUR (*Degree of Saturation*).

Durante a execução, o algoritmo demonstrou ser capaz de produzir uma coloração eficiente, utilizando 31 cores distintas, com tempo de processamento inferior a um segundo. A análise dos resultados não identificou colisões entre vértices adjacentes, evidenciando o correto funcionamento do algoritmo e sua adequação ao problema de alocação de frequências. Tais características confirmam o desempenho eficiente do DSATUR (*Degree of Saturation*) na geração de soluções próximas ao número cromático mínimo, com baixo custo computacional.

De acordo com **BRÉLAZ** (1979), o DSATUR (*Degree of Saturation*) se baseia na escolha do vértice com maior grau de saturação, ou seja, aquele com maior número de vizinhos já coloridos com cores distintas. Essa estratégia permite que o algoritmo produza soluções de alta qualidade sem a necessidade de técnicas exaustivas, sendo particularmente eficaz em grafos densos.

Em comparação com abordagens gulosas tradicionais, como os algoritmos First Fit e Welsh-Powell, o DSATUR (*Degree of Saturation*) apresenta desempenho superior. Estudo realizado por Aslan e **BAYKAN** (2016) mostrou que, embora algoritmos gulosos sejam mais simples e rápidos, tendem a utilizar um número maior de cores, enquanto o DSATUR (*Degree of Saturation*) consegue manter um equilíbrio mais consistente entre tempo de execução e qualidade da coloração.

Além disso, aplicações práticas em redes de comunicação reforçam a relevância do algoritmo. Conforme apresentado por **UYGUNGELLEN, AUER e BHARUCHA**(2011), o

DSATUR (*Degree of Saturation*) foi utilizado com sucesso em esquemas de reutilização dinâmica de frequência em redes de femtocélulas, garantindo a mitigação de interferências com baixa complexidade computacional. Esse tipo de aplicação evidencia a versatilidade do algoritmo para resolver problemas reais de alocação de recursos em sistemas de telecomunicações.

Ainda que existam algoritmos exatos, como o Branch-and-Cut (**Méndez-Díaz e Zabala**, 2006), capazes de encontrar a solução ótima, seu alto custo computacional os torna inviáveis para instâncias maiores. Nesse sentido, o DSATUR (*Degree of Saturation*) se mostra uma solução heurística eficiente e escalável, apropriada para contextos que exigem resposta rápida com boa aproximação do ótimo.

#### 4 CONCLUSÃO

A coloração de grafos é uma abordagem eficaz para resolver problemas de alocação de recursos, como a atribuição de frequências de rádio, onde o objetivo principal é evitar interferências. Através do desenvolvimento de um sistema baseado no algoritmo DSATUR (*Degree of Saturation*), foi possível demonstrar a aplicabilidade prática da coloração de grafos em contextos reais, garantindo soluções eficientes e com baixo custo computacional.

A utilização do DSATUR (*Degree of Saturation*) mostrou-se vantajosa em comparação a estratégias mais simples, como algoritmos puramente gulosos ou de backtracking, uma vez que equilibra qualidade de coloração e desempenho. Com base nos testes realizados, conclui-se que essa abordagem é promissora para aplicações de engenharia de comunicações, planejamento de redes e outras áreas que exigem soluções de coloração robustas.

#### 5 REFERÊNCIAS

- AIRES, Victória Patrícia Silva.** *Grafos rotulados, grafos graciosos e problemas de coloração especiais*. 2015. Relatório (Iniciação Científica) – Universidade Federal do Amazonas, Instituto de Computação, Manaus, 2015.
- NOGUEIRA JÚNIOR, Dârcio Costa.** *Grafos e problemas de caminhos*. 2017. Dissertação (Mestrado Profissional em Matemática em Rede Nacional) – Universidade Federal de Viçosa, Viçosa, 2017.
- BRÉLAZ, D.** New methods to color the vertices of a graph. *Communications of the ACM*, ACM, v. 22, n. 4, p. 251–256, 1979
- ASLAN, Murat; BAYKAN, Nurdan Akhan.** A performance comparison of graph coloring algorithms. *International Journal of Intelligent Systems and Applications in Engineering*, v. 4, p. 1–7, 2016.

**MÉNDEZ-DÍAZ, Isabel; ZABALA, Paula.** A Branch-and-Cut algorithm for graph coloring. *Discrete Applied Mathematics*, v. 154, n. 5, p. 826–847, 2006.

**UYGUNGELN, Serkan; AUER, Gunther; BHARUCHA, Zubin.** Graph-based dynamic frequency reuse in femtocell networks. In: *IEEE Vehicular Technology Conference (VTC)*. Anais... IEEE, 2011.