

# Programação Linear Inteira e Combinatória Poliédrica do Problema de Coloração em Grafos

Ieremies Vieira da Fonseca Romero

## **Resumo**

Histórico / importância do problema. Usar PLI, em especial branch-and-price (geração de colunas) e branch-and-cut (planos de corte), no problema de coloração de grafos. Objetivamos novas formulações, cortes e estratégias para melhor resolver as instâncias consolidadas da literatura.

## **1 Introdução**

Contexto histórico e aplicações práticas.

Importância de resolver o problema de forma eficaz e rápida.

### **1.1 Modelo matemático**

### **1.2 Revisão bibliográfica**

Focar na parte de PLI, não precisa trazer coisas de heurísticas ou outras técnicas.

## 2 Metodologia

**Programação Linear** é uma técnica de otimização de problemas a partir da modelagem dos mesmos em **programas lineares**. Nestes, definimos uma função objetivo, a qual queremos maximizar ou minimizar com suas variáveis sujeitas a um conjunto de restrições lineares (equações ou inequações lineares) [1]. Um programa linear pode ser escrito da seguinte forma:

$$\begin{aligned} \text{minimize} \quad & cx \\ \text{sujeito a} \quad & Ax \geq b \\ & x \in \mathbb{R}_+ \end{aligned}$$

Para encontrar soluções viáveis com valores ótimos, conhecemos o algoritmo **simplex** que, apesar de ter complexidade exponencial, no caso médio executa em tempo polinomial.

Para alguns problemas, como o de dominação romana, não faz sentido falar em soluções fracionárias, afinal, não conseguimos designar meia legião a uma cidade. Para isso, restringimos as variáveis aos inteiros, fazendo assim um **Programa Linear Inteiro**. Caso apenas um subconjunto das variáveis possuam a restrição de integralidade, chamamos esse programa de linear misto.

O que a princípio pode parecer uma pequena alteração, torna o problema computacionalmente ainda mais complexo. Para encontramos boas soluções viáveis para esse tipo de programa, algoritmos como o simplex não são o suficiente. Para isso, utilizamos técnicas como **branch-and-bound**,

que consiste em dividir o problema em subproblemas menores e, durante o processo, encontrar limitantes que permitam diminuir o espaço de busca.

Variações como **branch-and-cut**, na qual, ao atingir soluções não inteiras na relaxação linear usando o simplex, utilizamos algoritmos de plano de cortes para adicionar mais restrições até a solução fornecida pelo simplex na RL for inteira. Já para **branch-and-price**, essa técnica advém da observação que, para grandes problemas, grande parte das variáveis permanecem nulas entre as interações do *branch-and-bound*. Assim podemos inseri-las conforme progredimos na nossa busca utilizando técnicas de geração de colunas.

### 3 Objetivos

Neste projeto, objetivamos propor novos modelos de PLI para dominação romana e suas variantes explorando técnicas como *branch-and-cut* e *branch-and-price*. Além disso, estudaremos a possibilidade de novos cortes e limitantes para as formulações.

### 4 Cronograma

BEPE

Ensino de programação no elite

## 5 Material e método

Para o desenvolvimento do projeto, o aluno utilizará-se de artigos e materiais de consulta disponibilizados pela UNICAMP de maneira gratuita, grande parte desses de forma online ou por meio da Biblioteca do Instituto de Matemática, Estatística e Computação Científica.

Ademais, serão realizados encontros semanais entre o aluno e o orientador para debater os conteúdos estudados e acompanhar o progresso do projeto.

## 6 Avaliação dos resultados

Os algoritmos e modelos propostos serão comparados com as instâncias presentes na literatura, como em REFERÊNCIA e, caso necessário, novas instâncias poderão ser geradas.

Os resultados dos experimentos computacionais serão comparados utilizando técnicas como **Performance Profile** demonstrado por Dolan e Moré [2].

RELATÓRIOS

## Referências

- [1] Vasek Chvatal, Vaclav Chvatal e Vasek Chv?atal. *Linear Programming*. W. H. Freeman, 15 de set. de 1983. 500 pp. ISBN: 978-0-7167-1587-0. Google Books: DN20\\_tW\\_BV0C.

- [2] Elizabeth D. Dolan e Jorge J. Moré. “Benchmarking optimization software with performance profiles”. Em: *Mathematical Programming* 91.2 (1 de jan. de 2002), pp. 201–213. ISSN: 0025-5610, 1436-4646. DOI: 10.1007/s101070100263. (Acedido em 10/06/2022).