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

Dado um grafo não direcionado, o **problema de coloração de vértices** (do inglês, VCP) requer que designemos uma cor a cada vértice sem que este possua vizinhos de mesma cor e utilizemos o menor número possível de cores. Sua NP-completude foi demonstrada por Garey e Johnson [13].

Como discutido por [22], diversas são as aplicações do problema, como: agendamento [19], timetabling [31], alocação de resgistradores [6], comunicação de redes [5] e alocação de banda [12].

As aplicações acima deixam claro que encontrar uma boa (ou até ótima solução) para o problema é crucial. Cenários reais muitas vezes lidam com centenas de milhares de vértices, tornando necessário também resolvermos em tempo hábil.

isso ficou horrível

1.1 Modelo matemático

conceitos que são citados mais para frente (eu vou voltar aqui para escrever)

1.2 Revisão bibliográfica

eu não sei o que colocar aqui (ou se precisa existir essa parte) Todos os trabalhos que coloquei estão citado em outras partes do texto

1.3 Problemas similares

Cornaz e Jost [10] e Cornaz, Furini e Malaguti [9] usam técnicas de outros problemas para resolver o VCP.

1.3.1 Bandwidth Coloring Problem

O problema de **Multicoloração de Banda** (BMCP) é a combinação de dois problemas: Coloração de Banda e Multicoloração. No problema de **Coloração de Banda**, a diferença entre a cor de cada par de vértices adjacentes deve ser, ao menos, a distância entre os dois vértices. Assim, quando a disância é igual a 1, este problema é exatamente o VCP. No problema de **Multicoloração** um valor w_i é designado a cada vértice e w_i cores devem ser alocadas a este, de forma que um par qualquer de vértices adjacentes não compartilhe nenhum cor em comum.

Assim, no BMCP, é necessário atribuir, para um vértice i, w_i cores e para qualquer par de vértices adjacentes, cada combinação dois a dois de cores atribuidas a eles deve ter diferença maior que a distância entre os vértices.

Coloco a formulção de atribuição? Acho que não já que o objetivo é apenas mostrar que existem problemas similares

Este problema permite que situações mais complexas que o VCP sejam modeladas, como o problema de alocação de frequência em telecomunicações [1].

1.3.2 Bounded VCP

Muitas vezes, o recurso que queremos alocar é limitado. Assim, podemos colocar um peso w_i em cada vértice e limitar a soma dos pesos dos vértices alocados a cada uma das cores, uma restrição de capacidade. Este problema é conhecido como VCP Limitado ou Problema de Empacotamento com Conflito Connolly, Martello e Toth [8].

Seja C a capacidade de cada cor, a restrição de capacidade é dada por

$$\sum_{i=1}^{n} w_i x_{ih} \leq C, \quad h = 1, \ldots, n$$

e pode ser adicionada à formulação de ASS para modelar o problema de BVCP.

Isso tá antes da formulação

1.3.3 WVCP

1.3.4 Problemas similares

esse foi um que esbarrei. Eu vou atrás de outros como b-coloring (, equitable) coloring (, rainbow) coloring?

1. MSCP O problema de Coloração de Soma Mínima nossa função objetivo é soma dos inteiros (cores) alocados a cada vértice.

2 Metodologia

2.1 Programação Linear

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) Chvatal, Chvatal e Chv?atal [7] . Um programa linear pode ser escrito da seguinte forma:

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

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

2.2 Programação linear inteira

Para alguns problemas, como o de coloração de grafos, não faz sentido falar em soluções fracionárias, afinal, não conseguimos designar “meia cor” a um vértice. 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 muito 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.

2.3 Formulação clássica (atribuição)

Sabemos que n cores são suficientes para colorir um grafo G. Assim, podemos definir dois conjuntos de variáveis binárias: x_{ih} se o vértice i é colorido com a cor h e y_h se a cor h é utilizada. Dessa forma, definimos a seguinte formulação.

$$
\begin{array}{lllll}
\text{(ASS)} & \text{minimize} & \displaystyle\sum_{i=1}^{n} y_h & & \\
& \text{sujeito a} & \displaystyle\sum_{h=1}^{n} x_{ih} = 1 & & i \in V \\
& & x_{ih} + x_{jh} \leq y_h & & (i,j) \in E, h = 1, \ldots, n \\
& & x_{ih} \in \{0,1\} & & (i,j) \in E, h = 1, \ldots, n \\
& & y_i \in \{0,1\} & & i \in V
\end{array}
$$

Apesar de sua claridade e simplicidade, tal formulação vê pouca aplicação prática sem que apliquemos técnicas mais sofisticadas.

Esse fato se dá por dois motivos:

- Muitas soluções são simétricas umas às outras, já que as cores são indistiguíveis. Uma solução que utiliza k cores possui k permutações de cores do que é, efetivamente, a mesma solução.

- A relaxação linear do modelo é extremamente fraca.

Méndez-Dıaz e Zabala [25] e Méndez-Dıaz e Zabala [26] se dedicaram a resolver tais problemas. Méndez-Dıaz e Zabala [25] adicionaram a restrição

$$y_h \geq y_h + 1 \quad h = 1, \ldots, n-1$$

que garante que a cor $h+1$ só será utilizada se a cor h já estiver sendo.

Eles também acrescentaram diversas famílias de inequaldiades válidas ao politopo do novo modelo que são adicionadas ao algoritmo de *Branch-and-Cut* para fortalecer a relaxação linear além de implementar a estratégia de branching proposta por Brélaz [2] com resultados computacionais satisfatórios. definir

eu preciso mostrar as inqualidades?

Já Méndez-Dıaz e Zabala [26] apresentam mais duas variações da formulação ASS: uma onde a quantidade de vértices cuja cor $h+1$ é atribuida não pode ser maior que a quantidade atribuida a cor h e outro onde conjuntos independentes são ordenados pelo menor indice e apenas a cor h pode ser atribuída ao $h-$ésimo conjunto.

2.4 Formulação com representantes (campelo)

Campêlo, Corrêa e Frota [4] propõe uma formulação de representantes também utilizando essa idea de representantes para remover simetrias e Campêlo, Campos e Corrêa [3] incrementa nessa idea, avançando ainda mais o estudo poliédrico da formulação.

2.5 Formulação de cobertura de conjuntos (branch-and-price)

Proposto por Mehrotra e Trick [23], outra forma de entender o problema é imaginá-lo como um problema de cobertura de conjuntos onde os conjuntos disponíveis são os conjuntos independentes dos vértices.

conjunto conjunto conjunto

Assim, seja S a família de conjuntos impendentes do grafo G, a variável binária x_s é representa se o conjunto $s \in S$ está sendo usado ou não na solução. Nossa formulação então se dá por:

$$
\begin{aligned}
\text{minimize} & \sum_{s \in S} x_s \\
\text{sujeito a} & \sum_{s \in S : i \in s} x_s \geq 1 && i \in V \\
& y_s \in \{0, 1\} && s \in S
\end{aligned}
$$

Explicação das restrições?

Já essa formulação sofre de ter um número exponencial de variáveis.

Mehrotra e Trick [23] propôs um algoritmo de *branch-and-price* baseado na formulação de cobertura de conjuntos. O subproblema de geração de coluna caracteriza um **Problema de Conjunto Independente de peso**

definir

máximo.

$$
\begin{aligned}
\text{maximize} \sum_{i \in V} \pi_i z_i & & \\
\text{sujeito a} \quad z_i + z_j \leq 1 & \quad & (i,j) \in E \\
z_i \in \{0,1\} & \quad & i \in V
\end{aligned}
$$

onde z_i é uma variável binária que indica se o vértice i está incluso no conjunto independente e π_i é o valor ótimo da variável dual associado a restrição **??**. Tal problema pode ser resolvido de forma heurística para encontrar a coluna de custo reduzido com valor negativo. Em caso de soluções fráscionárias, os autores sugerem uma estratégia que garante que os subproblemas continuam a ser de coloração de vértices e apenas requer que o grafo original seja alterado.

preciso explica o porquê disso?

explico qual?

- Malaguti, Monaci e Toth [21] propôs metaheurísitcas para inicialização e geração de colunas bem como novos esquemas de branching.
- Held, Cook e Sewell [16] sugere técnicas para melhorar a estabilidade numérica

inline ainda preciso olhar esses dois papers

Hansen, Labbé e Schindl [15] propôs a formulação chamada de **Empa-**

cotamento de conjunto.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{s \in \Omega} (|s| - 1) x_s \\
\text{sujeito a} \quad & \sum_{s \in \Omega : i \in s} x_s \leq 1 \qquad i \in V \\
& y_s \in \{0, 1\} \quad s \in \Omega
\end{aligned}
$$

na qual Ω é a família de conjuntos independentes com mais de um elemento. Para essa formulação, seja z o valor da solução, a quantidade de cores usadas é igual $k = n - z$. Ale disso, Hansen, Labbé e Schindl [15] demonstram a equivalência das formulações de SC e SP, bem como apresentam diversas famílias de inequalidades válidas que definem facetas. definir

Hansen, Labbé e Schindl [15] Seja $i \in V$, então a inequação correspondente /ref{rest9} define uma faceta se, e somente se, i não for dominado.

definir dominado

Os autores também apresentam resultados computacionais que não demonstram superioridade entre o trabalho deles em relação à Mehrotra e Trick [23]. Por fim, duas técnicas de pré-processamento e um algoritmo de plano de corte . definir

2.6 Branch and bound usando DSATUR

Introduzido por Brélaz [2], melhorado por Sewell [29] e Segundo [28] e aplicado por Méndez-Dıaz, Nasini e Severın [24] em variantes do VCP.

Brélaz [2] propôs o algoritmo guloso chamado de DSATUR, em que, a cada iteração, colorimos um vértice v como uma cor válida . Dizemos que o definir

grau de saturação de um vértice v numa coloração parcial é a quantidade de cores distintas na sua vizinhança aberta . O DSATUR utiliza essa ideia para escolher, como próximo vértice a ser colorido, aquele com maior grau de saturação.

cromatico ou de saturação

definir

definir

É possível utilizar essa ideia para melhorar nosso *branch-and-bound*. A cada ramificação, selecionamos o vértice com maior gráu de saturação e criamos um problema para cada cor viável já utilizada, acrescentando uma ainda não utilizada.

talvez eu precise definir as notações de coloração parcial para isso ficar melhor

Apesar disso, muitas vezes, diversos vértices possuem o mesmo gráu de saturação, fazendo-se necessário implementar regras de desempate. Dentre as propostas, temos:

- Brélaz [2] utiliza o gráu do vértice.
- Sewell [29] utiliza o vértice que maximiza o número de cores disponíveis para todos os vértices ainda não coloridos.
- Segundo [28] incrementa na ideia anterior, mas apenas utilizando os vértices que estão sendo desempatados.

Em todos os casos acima, se mantiver algum empate, a ordenação lexigráfica é utilizada.

Ternier [30] implementa essas variações mostra que o proposto por Sewell [29] é o mais rápido, mesmo com maior complexidade computacional na regra de desempate, dado um bom limitante inferior inicial.

Ternier [30] apresenta novas variações para o algoritmo de *branch-and-bound* usando DSATUR e novas regras de escolha de vértices com bons resultados em relação ao estado-da-arte.

2.7 Ordenação parcial

ainda preciso estudar isso aqui

Jabrayilov e Mutzel [17] e Jabrayilov e Mutzel [18].

2.8 Estado da arte

Jabrayilov e Mutzel [17] implementam as abordagens acima e mostra não haver uma dominancia clara entre nenhuma delas. Apesar disso, nos seus testes, ordenação parcial se sai melhor em grafos esparços enquanto a formulação de representantes se sai melhor em grafos densos.

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.

aqui a minha ideia é apresentar esse tal de ferramental moderno e as ideias mais recentes que podemos aplicar

Lima, Iori e Miyazawa [20] Pessoa, Sadykov e Uchoa [27]

4 Cronograma

BEPE indicar umas possibilidades de nomes. Manuel Iori.

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 as *Graph Coloring Instances* [14] 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 [11].

RELATÓRIOS

Referências

[1] Karen I. Aardal et al. "Models and solution techniques for frequency assignment problems". Em: *Annals of Operations Research* 153.1 (mai. de 2007), pp. 79–129. DOI: `10.1007/s10479-007-0178-0`.

[2] Daniel Brélaz. "New methods to color the vertices of a graph". Em: *CACM* 22.4 (abr. de 1979), pp. 251–256. DOI: `10.1145/359094.359101`.

[3] Manoel Campêlo, Victor A. Campos e Ricardo C. Corrêa. "On the Asymmetric Representatives Formulation for the Vertex Coloring Problem". Em: *Discrete Applied Mathematics* 156.7 (abr. de 2008), pp. 1097–1111. ISSN: 0166-218X. DOI: `10.1016/j.dam.2007.05.058`. URL: `https://www-sciencedirect.ez88.periodicos.capes.gov.br/science/article/pii/S0166218X07002922` (acedido em 13/12/2022).

[4] Manoel Campêlo, Ricardo Corrêa e Yuri Frota. "Cliques, Holes and the Vertex Coloring Polytope". Em: *Information Processing Letters* 89.4 (fev. de 2004), pp. 159–164. ISSN: 0020-0190. DOI: `10.1016/j.ipl.2003.11.005`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S002001900300512X` (acedido em 13/12/2022).

[5] Alberto Caprara et al. "Passenger Railway Optimization". Em: *Discrete optimization / edited by K. Aardal, G.L. Nemhauser, R. Weismantel.* – (2007). DOI: `10.1016/S0927-0507(06)14003-7`.

[6] Fred C. Chow e John L. Hennessy. “The priority-based coloring approach to register allocation”. Em: *ACM Transactions on Programming Languages and Systems* (1990). DOI: `10.1145/88616.88621`.

[7] 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`.

[8] David Connolly, Silvano Martello e Paolo Toth. “Knapsack Problems: Algorithms and Computer Implementations”. Em: *The Journal of the Operational Research Society* 42.6 (jun. de 1991), p. 513. DOI: `10.2307/2583458`.

[9] Denis Cornaz, Fabio Furini e Enrico Malaguti. “Solving Vertex Coloring Problems as Maximum Weight Stable Set Problems”. Em: *Discrete Applied Mathematics* 217 (jan. de 2017), pp. 151–162. ISSN: 0166-218X. DOI: `10.1016/j.dam.2016.09.018`. URL: `https://www-sciencedirect.ez88.periodicos.capes.gov.br/science/article/pii/S0166218X16304103` (acedido em 13/12/2022).

[10] Denis Cornaz e Vincent Jost. “A one-to-one correspondence between colorings and stable sets”. Em: *Operations Research Letters* 36.6 (nov. de 2008), pp. 673–676. DOI: `10.1016/J.ORL.2008.08.002`.

[11] 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).

[12] A. Gamst. “Some lower bounds for a class of frequency assignment problems”. Em: *IEEE Transactions on Vehicular Technology* (1986). DOI: 10.1109/T-VT.1986.24063.

[13] Michael Randolph Garey e David S. Johnson. “Computers and Intractability: A Guide to the Theory of NP-Completeness”. Em: (1979).

[14] *Graph Coloring Instances*. URL: https://mat.tepper.cmu.edu/COLOR/instances.html (acedido em 14/12/2022).

[15] P. Hansen, M. Labbé e D. Schindl. “Set covering and packing formulations of graph coloring: Algorithms and first polyhedral results”. Em: *Discrete Optimization* 6.2 (mai. de 2009), pp. 135–147. DOI: 10.1016/j.disopt.2008.10.004.

[16] Stephan Held, William J. Cook e Edward C. Sewell. “Maximum-weight stable sets and safe lower bounds for graph coloring”. Em: *Mathematical Programming Computation* 4.4 (mai. de 2012), pp. 363–381. DOI: 10.1007/S12532-012-0042-3.

[17] Adalat Jabrayilov e Petra Mutzel. “New Integer Linear Programming Models for the Vertex Coloring Problem”. Em: *undefined* (2018). DOI: 10.1007/978-3-319-77404-6_47. URL: https://www.semanticscholar.org/reader/8bb7c1f678a354827560365ced3ae01a91c16a07 (acedido em 24/11/2022).

[18] Adalat Jabrayilov e Petra Mutzel. “Strengthened Partial-Ordering Based ILP Models for the Vertex Coloring Problem”. Em: *undefined* (2022). URL: https://www.semanticscholar.org/paper/Strengthened-

Partial-Ordering-Based-ILP-Models-for-Jabrayilov-Mutzel/4c2e06b3778930f6996467251e7135e57cfe1145 (acedido em 24/11/2022).

[19] F. Leighton. “A Graph Coloring Algorithm for Large Scheduling Problems.” Em: *Journal of research of the National Bureau of Standards* (1979). DOI: 10.6028/JRES.084.024.

[20] Vinıcius Loti de Lima, Manuel Iori e Flávio Keidi Miyazawa. “Exact solution of network flow models with strong relaxations”. Em: *Mathematical Programming* (mar. de 2022). DOI: 10.1007/s10107-022-01785-9.

[21] Enrico Malaguti, Michele Monaci e Paolo Toth. “An Exact Approach for the Vertex Coloring Problem”. Em: *Discrete Optimization* 8.2 (mai. de 2011), pp. 174–190. ISSN: 1572-5286. DOI: 10.1016/j.disopt.2010.07.005. URL: https://www-sciencedirect.ez88.periodicos.capes.gov.br/science/article/pii/S157252861000054X (acedido em 13/12/2022).

[22] Enrico Malaguti e Paolo Toth. “A Survey on Vertex Coloring Problems”. Em: *International Transactions in Operational Research* 17.1 (jan. de 2010), pp. 1–34. ISSN: 0969-6016. DOI: 10.1111/j.1475-3995.2009.00696.x. (Acedido em 13/12/2022).

[23] Anuj Mehrotra e Michael A. Trick. “A Column Generation Approach for Graph Coloring”. Em: *INFORMS Journal on Computing* 8.4 (nov. de 1996), pp. 344–354. ISSN: 1091-9856. DOI: 10.1287/ijoc.8.4.344. (Acedido em 13/12/2022).

[24] Isabel Méndez-Dıaz, Graciela Nasini e Daniel Severın. "A DSATUR-based algorithm for the Equitable Coloring Problem". Em: *Computers &amp Operations Research* 57 (mai. de 2015), pp. 41–50. DOI: 10.1016/j.cor.2014.11.014.

[25] Isabel Méndez-Dıaz e Paula Zabala. "A Branch-and-Cut Algorithm for Graph Coloring". Em: *Discrete Applied Mathematics* 154.5 (abr. de 2006), pp. 826–847. ISSN: 0166-218X. DOI: 10.1016/j.dam.2005.05.022. URL: https://www-sciencedirect.ez88.periodicos.capes.gov.br/science/article/pii/S0166218X05003094 (acedido em 13/12/2022).

[26] Isabel Méndez-Dıaz e Paula Zabala. "A Cutting Plane Algorithm for Graph Coloring". Em: *Discrete Applied Mathematics* 156.2 (jan. de 2008), pp. 159–179. ISSN: 0166-218X. DOI: 10.1016/j.dam.2006.07.010. URL: https://www-sciencedirect.ez88.periodicos.capes.gov.br/science/article/pii/S0166218X0700100X (acedido em 13/12/2022).

[27] Artur Pessoa, Ruslan Sadykov e Eduardo Uchoa. "Solving Bin Packing Problems Using VRPSolver Models". Em: *Operations Research Forum* 2.2 (abr. de 2021). DOI: 10.1007/s43069-020-00047-8.

[28] Pablo San Segundo. "A new DSATUR-based algorithm for exact vertex coloring". Em: *Computers Operations Research* 39.7 (jul. de 2012), pp. 1724–1733. DOI: 10.1016/J.COR.2011.10.008.

[29] E Sewell. "An improved algorithm for exact graph coloring". Em: *Cliques, Coloring, and Satisfiability.* American Mathematical Society, out. de 1996, pp. 359–373. DOI: 10.1090/dimacs/026/17.

[30] Ian-Christopher Ternier. "Exact Algorithms for the Vertex Coloring Problem and Its Generalisations". Tese de doutoramento. Université Paris sciences et lettres, 21 de nov. de 2017. URL: https://theses.hal.science/tel-01867961 (acedido em 04/01/2023).

[31] D. de Werra. "An introduction to timetabling". Em: *European Journal of Operational Research* (1985). DOI: 10.1016/0377-2217(85)90167-5.