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, uma **coloração própria** deste é uma alocação de cores tal que nenhum vértice seja adjacente a outro de mesma cor. O **problema de coloração de vértices** (do inglês, VCP) requer que encontremos uma coloração própria usando o menor número possível de cores. Sua NP-completude foi demonstrada por Garey e Johnson [15].

Como discutido por Malaguti e Toth [28], diversas são as aplicações do problema, como: agendamento [23], timetabling [36], alocação de registradores [6], comunicação de redes [5] e alocação de banda [14].

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 de forma eficiente.

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

Surveys do assunto:

- [28]
- [24]
- [20]
- [25]

1.3 Problemas similares

Cornaz e Jost [10] e Cornaz, Furini e Malaguti [9] usa stable set para resolver o VCP.

1.3.1 Bandwidth Coloring Problem

O problema de **Multicoloração de Banda** (do inglês, 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. 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 nenhuma 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 atribuídas a eles deve ter diferença maior que a distância entre os vértices. Este problema permite que situações mais complexas que o VCP sejam modeladas, como a alocação de frequência em telecomunicações [1].

1.3.2 Bounded VCP

Muitas vezes, o recurso que queremos alocar é limitado. Para refletir isso, podemos colocar um peso w_i em cada vértice e restringir a soma dos pesos dos vértices alocados a cada uma das cores, uma restrição de capacidade. Este problema é conhecido como VCP Limitado (do inglês, BVCP) ou **Problema de Empacotamento com Conflito** [8].

conflita com o w acima

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

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

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

BVCP.

Isso tá antes da formulação

1.3.3 WVCP

A cada vértice é atrelado um peso e o custo de uma cor é o maior peso dentre os vértices que ela colore. Assim, no **Problema de Coloração com Pesos**, queremos reduzir o custo total das cores. Esse problema vê aplicações na "SHEDULING ON A BATCH MACHINE WITH JOB COMPATIBILITIES" e Problema de Decomposição de Matriz em Divisão de tempo para Alocação de Tráfico de Multiplo Acessos (MATRIX DECOMPOSITION PROBLEM IN TIME DIVISION MULTIPLE ACCES TRAFFIC ASSIGNMENT). Esse problema foi estudado por [36], [12] e [13].

talvez uns resultados masi atualizados?

1.3.4 Problemas similares

- minimum sum
- equitable
- rainbow
- max-coloring
- minimun sum

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) [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.

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

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 indistinguíveis. 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 [30] e Méndez-Dıaz e Zabala [31] se dedicaram a resolver tais problemas. Méndez-Dıaz e Zabala [30] 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 inequalidades 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 definir branching proposta por Brélaz [2] com resultados computacionais satisfatórios.

Já Méndez-Dıaz e Zabala [31] apresentam mais duas variações da formulação ASS: uma onde a quantidade de vértices cuja cor $h+1$ é atribuída não pode ser maior que a quantidade atribuída a cor h e outro onde conjuntos independentes são ordenados pelo menor índice e apenas a cor h pode ser atribuída ao $h-$ésimo conjunto.

2.4 Formulação por representantes

Campêlo, Corrêa e Frota [4] propuseram uma formulação baseada em representantes, na qual cada cor é representada por um vértice. Para tal, utilizamos a variável binária x_{vu}, para todo $u, v \in V$ não adjacentes, a fim de representar se o vértice v é representante da cor de u e x_{vv} se v é o próprio representante de sua cor. Seja $\bar{N}(v)$ o conjunto de vértices não adjacentes de v, esta formulação pode ser escrita como

$$
\begin{aligned}
\text{(REP)} \quad & \text{minimize} \sum_{v \in V} xvv \\
& \text{sujeito a} \quad \sum_{u \in \bar{N}(v) \cup \{v\}} x_{uv} = 1 && v \in V \\
& \qquad x_{vu} + x_{vw} \leq x_{vv} && v \in V, \forall e = (u, w) \in G[\bar{N}(v)] \\
& \qquad x_{vu} \in \{0, 1\} && \text{para todo par de vértices } u, v \text{ não adjacentes ou } v = u
\end{aligned}
$$

O primeiro conjunto de restrições garante que todo vétice terá extamente um representante. O segundo garante que dois vértices adjacentes terão representantes diferentes.

Como Campêlo, Campos e Corrêa [3] discute, existem diversas soluções simétricas que apenas mudam o representante das cores sem alterar efetivamente a solução. Eles propõem acrescentar uma ordenação para que apenas o menor vértice podesse ser o representante. Porém, este modelo possui um número exponencial de variáveis. Os autores também apresentam diversas restrições válidas a fim de reforçar o modelo.

Por fim, Campêlo, Campos e Corrêa [3] debruçam-se sobre essa formulação, realizando a caracterização completa do politopo para algumas classes de grafos. Experimentos computacionais foram feitos por Jabrayilov e Mutzel [21] mostrando a capacidade deste modelo de competir com as demais formulações.

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

Proposto por Mehrotra e Trick [29], outra forma de entender o problema é imaginá-lo como um **problema de cobertura de conjuntos** (do inglês, SC) 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 representa se o conjunto $s \in S$ está sendo usado ou não na solução. Nossa formulação então se dá por:

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

O primeiro conjunto de restrições garante que todos os vértices de V estarão contido em algum conjunto indpendente escolhido. Apesar de poucas restrições, essa formulação sofre de ter um número exponencial de variáveis, o que a torna impossível de implementá-la em “SOLVERS” convencionais como *Gurobi*.

Por tal motivo, Mehrotra e Trick [29] propuseram um algoritmo de *branch-and-price* baseado nesta modelagem.

definir

acho que aqui tem que ir a definição de branch-and-price

Nest algoritmo, o subproblema de geração de coluna caracteriza um **Pro-**

blema de Conjunto Independente de peso máximo:

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

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 1. Tal problema pode ser resolvido de forma heurística para encontrar a coluna de custo reduzido com valor negativo. Em caso de soluções fracionárias, 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?

Além disso, Malaguti, Monaci e Toth [27] propôs meta-heurísticas para inicialização e geração de colunas bem como novos esquemas de branching. Já Held, Cook e Sewell [18] sugere técnicas para melhorar a estabilidade numérica

Similar a este último modelo, Hansen, Labbé e Schindl [17] propuseram a formulação chamada de **Empacotamento de conjunto** (do inglês, SP)

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

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 $k = n - z$. Além disso, Hansen, Labbé e Schindl [17] demonstram a equivalência das formulações de SC e SP, bem como apresentam diversas famílias de inequalidades válidas que definem facetas. definir

Preposição 2.1. *Hansen, Labbé e Schindl [17] 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 [29]. Por fim, duas técnicas de pré-processamento e um algoritmo de plano de corte . definir

2.6 Branch and bound usando DSATUR

Brélaz [2] propôs o algoritmo guloso chamado de DSATUR, em que, a cada iteração, colorimos um vértice v como uma cor válida . definir Dizemos que o **grau de saturação** cromatico ou de saturação de um vértice v numa coloração parcial é a quantidade de cores distintas na sua vizinhança aberta . definir O DSATUR utiliza essa ideia definir para escolher, como próximo vértice a ser colorido, aquele com maior grau de saturação.

É possível utilizar essa ideia para melhorar nosso *branch-and-bound*. A cada ramificação, selecionamos o vértice com maior grau 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 grau de saturação, fazendo-se necessário implementar regras de desempate. Dentre as propostas, temos:

- Brélaz [2] utiliza o grau do vértice.
- Sewell [34] utiliza o vértice que maximiza o número de cores disponíveis para todos os vértices ainda não coloridos.
- Segundo [33] 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 [35] implementa essas variações mostra que o proposto por Sewell [34], o qual se mostra o mais rápido, mesmo com maior complexidade computacional na regra de desempate, dado um bom limitante inferior inicial.

Ternier [35] 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 hibrida

Apresentado inicialmente por [21] e posteriormente melhorado por [22], utilizamos um misto de ordenação parcial da união entre os vértices e as cores disponíveis e o modelo de atribuição. Dizemos que o vértice v é colorido com a cor h se $h-1 \succ v$ e $h \not\succ v$ (no caso de $h=1$, se $h \not\succ v$). Além disso, nesse modelo, é escolhido um vértice arbitrário q. A formulação segue:

definir H como upper bound

$$
\begin{aligned}
(POPH) \quad & \text{minimize } 1 + \sum_{1 \le h \le H} g_{h,q} \\
& \text{sujeito a} \quad g_{H,v} = 0 && \forall v \in V \\
& x_{v,1} = 1 - g_{1,v} && \forall v \in V \\
& x_{v,h} = g_{h-1,v} - g_{h,v} && \forall v \in V, h = 2, \ldots, H \\
& x_{u,1} + x_{v,1} \le g_{1,q} && \forall uv \in E \\
& x_{u,h} + x_{v,h} \le g_{h-1,q} && \forall uv \in E, h = 2, \ldots, H \\
& g_{h,q} - g_{h,v} \ge 0 && \forall v \in V, h = 1, \ldots, H \\
& g_{h+1,q} - g_{h,v} \ge 0 && \forall v \in N(q), h = 1, \ldots, H-1 \\
& x_{v,h}, g_{h,v} \in \{0,1\} && \forall v \in V, h = 1, \ldots, H.
\end{aligned}
$$

vou reformatar essa explicação

O primeiro conjunto de restrições garante que nenhum vértice é maior que a cor H. O segundo e terceiro correlacionam as variáveis de ordenação parcial com as de atribuição. O quarto e quinto garantem que dois vértices adjacentes não são coloridos com a mesma cor. Já a sexta, força que q seja o vértice com a maior cor. A sétima é utilizada para reforçar.

Segundo os resultados experimentais de [22], essa formulação domina os modelos anteriores nas instâncias DIMACS [16] esparsas (densidade $\frac{2|E|}{|V|(|V|-1)} \le$

0.1).

2.8 Diagrama de decisões binárias ordenadas

[19]

2.9 Estado da arte

Jabrayilov e Mutzel [21] implementam as abordagens acima e mostra não haver uma dominância clara entre nenhuma delas. Apesar disso, nos seus testes, ordenação parcial se sai melhor em grafos esparsos 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 [26] Pessoa, Sadykov e Uchoa [32]

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* [16] 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] Vašek Chvátal. *Linear Programming.* W. H. Freeman, 1983, p. 478. 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] Bruno Escoffier, Jérôme Monnot e Vangelis Th. Paschos. “Weighted Coloring: further complexity and approximability results”. Em: *Information Processing Letters* 97.3 (fev. de 2006), pp. 98–103. DOI: 10.1016/j.ipl.2005.09.013.

[13] Gerd Finke et al. “Batch processing with interval graph compatibilities between tasks”. Em: *Discrete Applied Mathematics* 156.5 (mar. de 2008), pp. 556–568. DOI: 10.1016/j.dam.2006.03.039.

[14] 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.

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

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

[17] 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.

[18] 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.

[19] Willem-Jan van Hoeve. “Graph coloring with decision diagrams”. Em: *Mathematical Programming* 192.1-2 (mai. de 2021), pp. 631–674. DOI: 10.1007/s10107-021-01662-x.

[20] Thore Husfeldt. *Graph colouring algorithms.* 2015. DOI: 10.48550/ARXIV.1505.05825.

[21] 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).

[22] 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).

[23] 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.

[24] R. M. R. Lewis. *Guide to Graph Colouring Algorithms and Applications. Algorithms and Applications*. Springer London, Limited, 2015, p. 253. ISBN: 9783319257303.

[25] Alane Marie De Lima e Renato Carmo. “Exact Algorithms for the Graph Coloring Problem”. Em: *Revista de Informática Teórica e Aplicada* 25.4 (nov. de 2018), p. 57. DOI: 10.22456/2175-2745.80721.

[26] 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.

[27] 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).

[28] 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).

[29] 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).

[30] 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).

[31] 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).

[32] 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.

[33] 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.

[34] 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.

[35] 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).

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