Programação Linear Inteira aplicada a problemas de Coloração em Grafos

Ieremies Vieira da Fonseca Romero

Resumo

Uma coloração própria de vértices de um grafo consiste em uma alocação de cores tal que nenhum par de vértices adjacentes possuam a mesma cor. No problema de coloração de vértices, dado um grafo, procuramos uma coloração própria utilizando o menor número possível de cores. Além de ser um tradicional problema de teoria dos grafos, o problema se apresenta como NP-difícil e resolvê-lo de forma exata é extremamente relevante na atualidade. Tais aplicações vão desda alocação de recursos como registradores até o gerenciamento de eventos.

Soluções exatas para este problema utilizam técnicas como enumeração com *Branch-and-Bound* [19], programação dinâmica [15, 5] ou programação linear inteira. Nesse projeto, almejamos aplicar a última e utilizar algoritmos de *Branch-and-Price* para superar os resultados da literatura.

Nosso trabalho se diferencia ao aplicar técnicas recentes em *branch-and-price* [43, 34] cuja eficácia já foi demonstrada em outros problemas, o que pode levar a uma melhora do estado-da-arte. Dessa forma, objetivamos resultados competitivos nas instâncias consolidadas na literatura, bem como possíveis novas formulações, cortes e estratégias para melhor resolvê-las.

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* (VCP) requer que encontremos uma coloração própria usando o menor número possível de cores. Decidir se um grafo possui uma coloração própria que usa até $k \geq 3$ cores é um problema NP-completo [21], de forma que o problema de otimização também é NP-difícil.

Muitas vezes, desejamos alocar eventos ou recursos sem que hajam conflitos, seja na divisão de tarefas num cenário de fábricas ou escritórios, ou na organização de eventos. Assim, desejamos construir uma alocação desses a horários sem que, por exemplo, requeira que uma pessoa esteja em dois locais simultaneamente. Tal problema pode ser convertido a um de coloração transformando cada evento a um vértice no qual dois vértices estão ligados se há necessidade de ao menos uma pessoa estar em ambos eventos. Em uma solução, cada cor irá representar um horário para os eventos acontecerem.

Além dessa, como discutido por Malaguti e Toth [36], diversas são as aplicações do problema, como: agendamento [31], tabela de horários [45], alocação de registradores [9], comunicação de redes [8] e alocação de banda [20].

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 rápida.

Neste projeto, utilizaremos de Programação Linear Inteira para resolver o problema de forma exata, como discutido na Seção 2. É importante mencionar que Programação Dinâmica também é comumente utilizada [16, 5], mas que vê apenas desenvolvimentos teóricos sem grande aplicabilidade prática no VCP. Já técnicas de enumeração baseado em *branch-and-bound* [4] possuem resultados práticos melhores que programação dinâmica, mas ainda são superados por técnicas de programação linear inteira. Recentemente, observamos avanços significativos utilizando técnicas de enumeração e *SAT* por Hebrard e Katsirelos [23].

Por fim, recomendamos as seguintes revisões bibliográficas sobre o problema [36, 32, 27, 33].

1.1 Modelo matemático

Durante essa proposta, utilizaremos algumas notações que apresentaremos a seguir.

Seja $G = (V, E)$ um grafo simples não direcionado, denotamos como $n = |V|$ a quantidade de vértices e $m = |E|$ a quantidade de arestas. Dizemos que a vizinhança aberta $N(v)$ de um vértice $v \in V$ no grafo G é o conjunto de vértices adjacentes a v. Já a vizinhança fechada $N[v] = N(v) \cup \{v\}$ é a vizinhança aberta adicionado do vértice v.

Muitas vezes, nos referimos a uma coloração própria apenas como coloração. Já uma coloração parcial é a atribuição, segundo as mesmas regras, de uma parte dos vértices e, em geral, utilizaremos essa definição para soluções em construção. O número cromático de G, chamado de $\chi(G)$, é a menor quantidade de cores que pode constituir uma coloração do grafo G. Por fim, usaremos H como limitante superior ao número cromático. Uma cor h é dita válida para o vértice v se ela não foi atribuída a nenhum vértice da vizinhança aberta de v.

Um conjunto $S \subseteq V$ é dito independente em G se, e somente se, não existem arestas em E que conectem dois elementos de S. Perceba que o conjunto de vértices coloridos com mesma cor deve ser um conjunto independente.

1.2 Problemas similares

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. No problema de *Multicoloração*, uma certa quantidade de cores devem ser alocadas a cada vértice, de forma que um par qualquer de vértices adjacentes não possuam uma cor em comum.

Assim, no BMCP, é necessário atribuir, para cada vértice, uma certa quantidade de 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].

Por outro lado, muitas vezes, o recurso que queremos alocar é limitado. Para refletir isso, podemos colocar um peso 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 (BVCP) ou *Problema de Empacotamento com Conflito* [11]

Tal restrição de capacidade pode ser adicionada à formulação de atribuição (ASS) para modelar o problema de BVCP sem grandes dificuldades.

Por fim, podemos atrelar a cada vértice, um peso e definir o custo de uma cor como 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 *Alocação de Máquinas com Compatibilidade de Tarefas* e *Problema de Decomposição de Matriz em Divisão de tempo para Alocação de Tráfico de Múltiplo Acessos* [45, 17, 18].

Além das generalizações apresentadas, existem diversas variações desse problema. A variante de *soma mínima* pede para encontramos a coloração com a menor soma de valores associados aos vértices [29]. Já a *equitativa,* é o desafio de encontrar uma coloração que distribua as cores de forma onde a diferença entre a cardinalidade da cor atribuída a dois vértices adjacentes seja pequena [40].

2 Metodologia

A seguir, apresentamos as técnicas que almejamos utilizar bem como abordagens já presentes na literatura disponível.

2.1 Programação Linear Inteira

Programação Linear é uma técnica de otimização a partir da modelagem de *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) [10]. 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 [3]. Além deste, é importante mencionar os métodos dos pontos interiores [30] e elipsoide [2] como alternativas.

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, tornando-o 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.

Por fim, é interessante nesse momento traçar o paralelo com a *Combinatória Poliédrica*. Definimos um *poliedro* como um subconjunto $P \subseteq \mathbb{R}^n$ tal que ele possa ser descrito por meio

de inequações lineares em tal espaço, similar aos programas lineares apresentados acima. Caso esse poliedro seja limitado, o chamamos de *politopo*.

Dizemos que um subconjunto do poliedro definido por uma inequação é uma *face*. Se esta não for igual ao poliedro inteiro, dizemos ser própria e, caso também não seja vazia, dizemos ser não-trivial. Uma *faceta* pode ser definida como uma face própria maximal, ou seja, uma face própria que não está contida em nenhuma outra.

A seguir, apresentamos as metodologias e técnicas que nos são relevantes para resolver o problema de forma exata. Em geral, as instâncias usadas para os resultados de experimentos computacionais são as apresentadas por DIMACS [13], que foram atualizadas com o passar dos anos.

2.2 Atribuição

É possível perceber que n cores são suficientes para colorir um grafo G. Podemos então 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, construímos a seguinte formulação.

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

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:

1. Muitas soluções são simétricas umas às outras, já que as cores são indistinguíveis. Uma solução que utiliza k cores possui $\frac{n!}{(n-k)!}$ permutações de cores do que é, efetivamente, a mesma solução.

2. A relaxação linear do modelo (quando a restrição de integridade é removida) é extremamente fraca. Assim, resolvê-la ainda nos deixa muito distante da solução ótima inteira.

De modo a resolver tais problemas, Méndez-Díaz e Zabala [38] 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, quebrando várias simetrias.

Eles também acrescentaram diversas famílias de desigualdades 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 ramificação proposta por Brélaz [4] com resultados computacionais satisfatórios.

Um algoritmo de *branch-and-cut* consiste em, durante a árvore de *branch-and-bound*, adicionar desigualdades e cortes válidos para diminuir o espaço de busca. Um corte válido é uma inequação que não elimina nenhuma solução inteira viável do modelo.

Já Méndez-Díaz e Zabala [39] 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.

2.3 Representantes

Campêlo, Corrêa e Frota [7] propuseram uma formulação baseada em representantes, em que cada cor é representada por um vértice. Para tal, utilizam 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

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

O primeiro conjunto de restrições garante que todo vértice terá exatamente um representante enquanto o segundo garante que dois vértices adjacentes terão representantes diferentes.

Como Campêlo, Campos e Corrêa [6] discutem, existem diversas soluções simétricas para tal modelo que apenas distinguem no representante das cores sem alterar efetivamente a solução. Eles propõem acrescentar uma ordenação para que apenas o menor vértice possa ser o representante, porém esta versão possui um número exponencial de variáveis e, portanto, requer técnicas como *branch-and-price*, abordada mais afundo na Seção 2.4. Os autores também apresentam diversas restrições válidas a fim de reforçar o modelo.

Experimentos computacionais foram feitos por Jabrayilov e Mutzel [28] mostrando a capacidade deste modelo de competir com as demais formulações.

2.4 Conjuntos independentes

Proposto por Mehrotra e Trick [37], outra forma de entender o problema é imaginá-lo como um *problema de cobertura de conjuntos* (SC) onde os conjuntos disponíveis são os conjuntos independentes dos vértices. Nesta classe de problemas, desejamos escolher o menor número de conjuntos tais que todos os elementos apareçam ao menos uma vez em cada um deles.

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:

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

O primeiro conjunto de restrições garante que todos os vértices de V estão contidos em algum conjunto independente escolhido. Apesar de poucas restrições, essa formulação sofre de ter um número exponencial de variáveis, o que a torna impraticável de implementá-la em resolvedores convencionais como *Gurobi*.

Por tal motivo, Mehrotra e Trick [37] propuseram um algoritmo de *branch-and-price* baseado nesta modelagem. Essa abordagem permite lidar com um número exponencial de variáveis, já que iniciamos com um subconjunto destas e adicionamos as necessárias conforme as iterações.

Neste algoritmo, a decisão de qual variável será adicionada é feita pelo subproblema de geração de coluna caracterizado por um *Problema de Conjunto Independente de Peso*

Máximo:

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

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 à restrição (1). Tal problema pode ser resolvido de forma heurística para encontrar a coluna de custo reduzido com valor negativo ou por técnicas como a proposta por Held, Cook e Sewell [24]. Em caso de soluções fracionárias, Mehrotra e Trick [37] sugerem uma estratégia que garante que os subproblemas continuam a ser de coloração de vértices e apenas requer que o grafo original seja alterado.

Além disso, Malaguti, Monaci e Toth [35] propõem meta-heurísticas para inicialização e geração de colunas bem como novos esquemas de ramificação.

Similar a este último modelo, Hansen, Labbé e Schindl [22] propuseram a formulação chamada de *Empacotamento de conjunto* (SP)

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

em que 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 [22] demonstram a equivalência das formulações de SC e SP, bem como apresentam diversas famílias de inequalidades válidas que definem facetas.

Os autores também apresentam resultados computacionais que não demonstram superioridade entre o trabalho deles em relação à Mehrotra e Trick [37] bem como técnicas de pré-processamento e um algoritmo de planos-de-corte. Um algoritmo de *planos-de-corte* é aquele que parte de um modelo relaxado, resolve-se e iterativamente acrescenta-se inequações válidas, chamadas de planos-de-corte, para eliminar tal solução. Repetimos o processo até encontrar uma solução para o problema original.

Já Morrison et al. [42] apresentam outra forma de realizar a ramificação do processo de *branch-and-bound*. Tradicionalmente, esta é feita dividindo apenas em dois subproblemas, mas os autores propõem uma abrangente quantidade de subproblemas a cada ramificação com intuito de impedir que cheguemos a subgrafos onde o problema de precificação é mais difícil. Mais uma vez, resultados computacionais provam que a ideia é competitiva com as anteriores.

Recentemente, Morrison, Sewell e Jacobson [41] propuseram uma estratégia utilizando diagramas de decisão para agilizar a resolução do problema de precificação que, segundo seus experimentos, possui bons resultados práticos bem como fornece limites inferiores durante o processo.

2.5 Ordenação parcial hibrida

Apresentado inicialmente por Jabrayilov e Mutzel [28], essa formulação utiliza um misto do modelo de atribuição e a ordenação parcial. Para isto, definimos uma ordenação parcial da união do conjunto de vértices e do conjunto ordenado de cores $(1, \ldots, H)$ e, portanto, determinamos a ordem relativa de cada vértice com respeito a cada cor. 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 e H é um limitante superior

do número cromático. A formulação segue:

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

O primeiro conjunto de restrições garante que nenhum vértice é maior que a cor H. Já o segundo e terceiro correlacionam as variáveis de ordenação parcial com as de atribuição, enquanto o quarto e quinto garantem que dois vértices adjacentes não são coloridos com a mesma cor. Por fim o sexto, força que q seja o vértice com a maior cor que, juntamente com o sétimo, são utilizados para reforçar a formulação.

Como demonstrado pelo autor, essa formulação apresenta-se também competitiva com as demais.

2.6 Diagrama de decisões binárias ordenadas

Por fim, mais recentemente, Hoeve [25] aplica técnicas de diagrama de decisão. Nesta, representamos cada atribuição por um arco que liga o "estado da solução" antes e depois de tomá-la. Mais formalmente, para um problema P definido por um conjunto de variáveis

ordenadas $X = \{x_1, x_2, \ldots, x_n\}$, construímos um diagrama de decisão, um grafo simples acíclico de $n+1$ níveis. O primeiro nível possui apenas um vértice, r, chamado raiz, assim como o último com o vértice t. O nível i é um conjunto de nós associados à variável x_i onde cada um destes possuem arcos para vértices do nível $i+1$ e tais arcos possuem etiquetas, 0 ou 1, correspondendo da variável associada.

Hoeve [25] utiliza a formulação de conjuntos independentes para aplicar tal técnica, onde cada variável indica o uso, ou não, de um vértice em um conjunto. Se conseguíssemos achar um diagrama de decisão que correspondesse exatamente ao problema de coloração, poderíamos resolver o problema por meio de uma formulação de fluxo, na qual cada $\{r,t\}-$caminho seria um conjunto independente e, como demostrado pelos autores, a função objetivo seria nosso número cromático. Infelizmente, tal diagrama pode conter um número exponencial de nós, requerendo técnicas mais sofisticadas. Além disso, quando a solução não é exata, a qualidade da solução depende da ordenação das variáveis.

Por exemplo, observe a Figura 1. Ao lado esquerdo está descrito um grafo e ao lado direito, um correspondente diagrama de decisão. Neste, os números dentro dos nós representam o conjunto de vértices disponíveis, ou seja, o estado, enquanto os arcos tracejados correspondem aos 0-arcos e os contínuos, os 1-arcos.

Os autores relatam ter obtido resultados competitivos com outros estados-da-arte utilizando estratégias específicas da literatura de diagrama de decisões para resolver as dificuldades apresentadas acima.

3 Objetivos

Recentemente, houve avanços significativos na abordagem de branch-and-price, resultando em vários trabalhos publicados para outros problemas que utilizam e melhoram essa técnica. Um exemplo notável é o trabalho de Lima, Iori e Miyazawa [34], que apresenta técnicas fortes que melhoram o estado da arte para problemas como o Problema de Empacotamento.

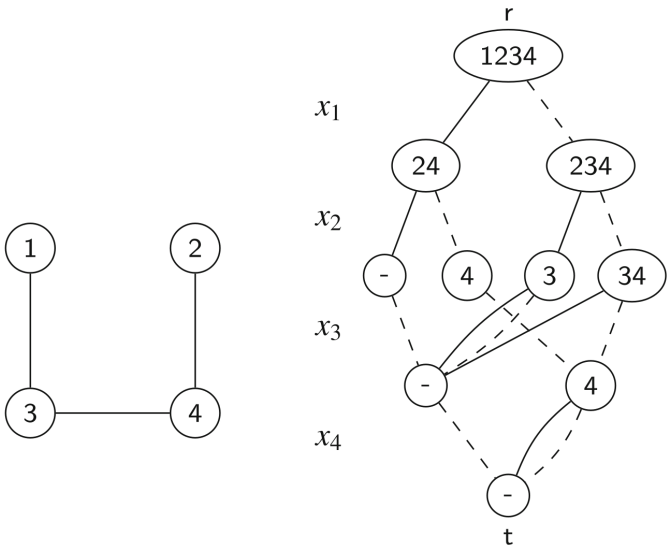

Figura 1: Imagem original por Hoeve [25]. Ao lado esquerdo, um grafo, e ao lado direito, um correspondente diagrama de decisão.

Os autores propõem um modelo baseado em fluxo de arcos para auxiliar em algoritmos de geração de colunas. Eles também comentam que qualquer problema de cobertura de conjunto consegue ser transformado em um problema de fluxo, o que indica que suas técnicas podem ser relevantes para o nosso problema. Além disso, eles utilizam *fixação de variáveis*, técnica na qual é possível provar que algumas variáveis nunca poderão entrar no modelo de *branch-and-pricing* e tornar a solução melhor.

A dificuldade reside em encontrar uma solução dual viável que possibilite computar o custo reduzido, necessário para provar esta afirmação. Um importante desenvolvimento proposto pelos autores é justamente um modelo linear capaz de encontrar eficientemente uma solução, mesmo que não seja ótima (o que, como argumentado por eles, é ainda mais eficiente).

Outro bom indicador do que pretendemos fazer é a semelhança de bons resultados recentes como Hoeve [25] que utilizam ideias muito similares.

Técnicas interessantes também foram propostas por Pessoa, Sadykov e Uchoa [43] que

apresentam um modelo genérico para resolver problema de roteamento. Quando se adiciona um corte no *branch-cut-and-price*, isso corresponde a variáveis no dual dificultando o subproblema de geração de colunas. Os autores utilizam cortes de rank-1 com memória limitada para melhorar tal processo. Além disso, eles usam *propagação de etiquetas*, técnica comum na resolução de problemas de precificação que pode ser interessante para o nosso problema.

Tendo em vista que tais técnicas ainda não foram aplicadas ao problema de coloração, objetivamos principalmente aplicarmo-nas tais bem como desenvolver qualquer necessidade de modificação ou melhoria. Além disso, consideramos estudar a adição de algoritmos de planos de corte, algo que já foi estudado brevemente na literatura, mas que ainda não viu grande impacto prático. Schindl, David [44] introduziu alguns resultados poliédricos sobre a formulação de cobertura de conjuntos, Hansen, Labbé e Schindl [22] tentaram aplicar alguns cortes (sem muito sucesso) mas Hulst [26] conseguiu avanços significativos em sua tese.

Por fim, como apresentado no começo desse projeto, diversos são os problemas similares ou generalizações bem como as aplicações de tais. Assim, podemos também voltar nossas técnicas e implementações a tais variantes.

4 Cronograma

O projeto está dividido em etapas distintas, cada uma com um objetivo específico. A primeira fase é de revisão bibliográfica, onde serão dedicados 2 trimestres para estudar a literatura existente sobre o assunto. Reforçamos que, apesar deste ser o foco da primeira etapa, a revisão da bibliografia será permanente. Em seguida, serão dedicados 2 trimestres para implementar o estado da arte, onde aplicaremos os conhecimentos adquiridos na fase anterior. A próxima etapa é o desenvolvimento de novos algoritmos, bem como a implementação e experimentação de tais soluções, onde serão dedicados os trimestres seguintes. Por último, nos dedicaremos a escrever o relatório final, que apresentará as conclusões e resultados do projeto.

Além disso, é importante notar que é de interesse do aluno realizar um intercâmbio com pesquisadores do exterior, como Manuel Iori. Pesquisadores como ele possuem vasta experiência tanto no problema quanto na técnica que estudamos e acreditamos que será engrandecedor para o processo de formação do aluno. Pretendemos concretizar tal intercâmbio no primeiro semestre de 2024.

Por fim, o aluno irá, como requerido pela instituição, cursar as três matérias de pós-graduação restantes no primeiro ano, defender seu exame de qualificação de mestrado (EQM) na metade de 2023 e realizar o Programa de Estágio em Docência (PED) no segundo semestre de 2023.

Refletimos o cronograma planejado na Tabela 1.

Tabela 1: Cronograma trimestral para este projeto de mestrado, iniciando em Março de 2023.

Atividade	Mar	Jun	Set	Dez	Mar	Jun	Set	Dez
Disciplinas	•	•						
EQM		•						
PED			•					
Revi. Biblio.	•	•	•					
Implem. estado-da-arte			•	•				
Desenvolvimento			•	•	•	•	•	•
Escrita							•	•

Acrescido a isso, o beneficiário deste projeto ministra aulas de Introdução à Programação para alunos de ensino médio na modalidade de itinerários formativos desde 2022 em colégio particular de Campinas, SP. Para o ano de 2023, estão programadas 3 aulas semanais, o que se mantém consoante a Portaria PR n° 05/2012. Acreditamos que tal atividade muito acrescentará no desenvolvimento do aluno como acadêmico, transmitindo os conhecimentos adquiridos à comunidade e ajudando na divulgação científica.

5 Material e método

Para o desenvolvimento do projeto, o aluno utilizará 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 instâncias do DIMACS [13] e, caso necessário, novas instâncias poderão ser geradas.

Os resultados dos experimentos computacionais serão comparados utilizando técnicas como *Performance Profile* apresentado por Dolan e Moré [14].

Além disso, pretendemos utilizar técnicas de análise estatística como apresentadas por Derrac et al. [12] a fim de produzir comparações mais rigorosas entre as propostas desse projeto.

Por fim, como previsto, serão elaborados relatórios com os resultados obtidos bem como qualquer artigo que seja relevante.

Referências

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

[2] R. G. Bland, D. Goldfarb e M. J. Todd. "Feature Article—The Ellipsoid Method: A Survey". *Operations Research* 29.6 (1981), pp. 1039–1091. DOI: 10.1287/opre.29.6.1039.

[3] K. H. Borgwardt. *The Simplex Method - Algorithms and Combinatorics*. Springer, 1986. ISBN: 9783540170969.

[4] D. Brélaz. "New methods to color the vertices of a graph". *Communications of the ACM* 22.4 (1979), pp. 251–256. DOI: 10.1145/359094.359101.

[5] J. M. Byskov. "Chromatic Number in Time $O(2.4023^n)$ Using Maximal Independent Sets". *BRICS Report Series* 9.45 (2002). DOI: 10.7146/brics.v9i45.21760.

[6] M. Campêlo, V. A. Campos e R. C. Corrêa. "On the Asymmetric Representatives Formulation for the Vertex Coloring Problem". *Discrete Applied Mathematics* 156.7 (2008), pp. 1097–1111. ISSN: 0166-218X. DOI: 10.1016/j.dam.2007.05.058.

[7] M. Campêlo, R. Corrêa e Y. Frota. "Cliques, Holes and the Vertex Coloring Polytope". *Information Processing Letters* 89.4 (2004), pp. 159–164. ISSN: 0020-0190. DOI: 10.1016/j.ipl.2003.11.005.

[8] A. Caprara et al. "Passenger Railway Optimization". *Discrete optimization* 14 (2007), pp. 129–187. DOI: 10.1016/S0927-0507(06)14003-7.

[9] F. C. Chow e J. L. Hennessy. "The priority-based coloring approach to register allocation". *ACM Transactions on Programming Languages and Systems* 12.4 (1990), pp. 501–536. DOI: 10.1145/88616.88621.

[10] V. Chvátal. *Linear Programming*. W. H. Freeman, 1983. ISBN: 978-0-7167-1587-0.

[11] D. Connolly, S. Martello e P. Toth. *Knapsack Problems: Algorithms and Computer Implementations. algorithms and computer implementations*. Vol. 42. 6. J. Wiley Sons, 1990, p. 296. ISBN: 0471924202. DOI: 10.2307/2583458.

[12] J. Derrac et al. "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms". *Swarm and Evolutionary Computation* 1.1 (2011), pp. 3–18. DOI: 10.1016/j.swevo.2011.02.002.

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

[14] E. D. Dolan e J. J. Moré. "Benchmarking optimization software with performance profiles". *Mathematical Programming* 91.2 (2002), pp. 201–213. DOI: 10.1007/s101070100263.

[15] D. Eppstein. "Small Maximal Independent Sets and Faster Exact Graph Coloring". *Journal of Graph Algorithms and Applications* 7.2 (2003), pp. 131–140. DOI: 10.7155/jgaa.00064.

[16] D. Eppstein. "Small Maximal Independent Sets and Faster Exact Graph Coloring". Em: *Graph Algorithms and Applications 4*. World Scientific, 2006, pp. 131–140. DOI: 10.1142/9789812773296_0006.

[17] B. Escoffier, J. Monnot e V. T. Paschos. "Weighted Coloring: further complexity and approximability results". *Information Processing Letters* 97.3 (2006), pp. 98–103. DOI: 10.1016/j.ipl.2005.09.013.

[18] G. Finke et al. "Batch processing with interval graph compatibilities between tasks". *Discrete Applied Mathematics* 156.5 (2008), pp. 556–568. DOI: 10.1016/j.dam.2006.03.039.

[19] F. Furini, V. Gabrel e I.-C. Ternier. "An Improved DSATUR-Based Branch-and-Bound Algorithm for the Vertex Coloring Problem". *Networks* 69.1 (2017), pp. 124–141. ISSN: 0028-3045. DOI: 10.1002/net.21716.

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

[21] M. R. Garey e D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness. a guide to the theory of NP-completeness*. W. H. Freeman, 1979, p. 338. ISBN: 0716710447.

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

[23] E. Hebrard e G. Katsirelos. "Constraint and Satisfiability Reasoning for Graph Coloring". *Journal of Artificial Intelligence Research* 69 (2020), pp. 33–65. ISSN: 1076-9757. DOI: 10.1613/jair.1.11313.

[24] S. Held, W. J. Cook e E. C. Sewell. "Maximum-weight stable sets and safe lower bounds for graph coloring". *Mathematical Programming Computation* 4.4 (2012), pp. 363–381. DOI: 10.1007/S12532-012-0042-3.

[25] W.-J. van Hoeve. "Graph coloring with decision diagrams". *Mathematical Programming* 192.1-2 (2021), pp. 631–674. DOI: 10.1007/s10107-021-01662-x.

[26] R. v. d. Hulst. "A branch-price-and-cut algorithm for graph coloring". Diss. de mestr. University of Twente, 2021. URL: http://essay.utwente.nl/88263/.

[27] T. Husfeldt. *Graph colouring algorithms*. 2015. DOI: 10.48550/ARXIV.1505.05825.

[28] A. Jabrayilov e P. Mutzel. "New Integer Linear Programming Models for the Vertex Coloring Problem". Em: *LATIN 2018: Theoretical Informatics*. 2018, pp. 640–652. DOI: 10.1007/978-3-319-77404-6_47.

[29] Y. Jin, J.-P. Hamiez e J.-K. Hao. "Algorithms for the minimum sum coloring problem: a review". *Artificial Intelligence Review* 47.3 (2016), pp. 367–394. DOI: 10.1007/s10462-016-9485-7.

[30] N. Karmarkar. "A new polynomial-time algorithm for linear programming". Em: *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM Press, 1984. DOI: 10.1145/800057.808695.

[31] F. Leighton. "A Graph Coloring Algorithm for Large Scheduling Problems." *Journal of Research of the National Bureau of Standards* 84.6 (1979), pp. 489–506. DOI: 10.6028/JRES.084.024.

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

[33] A. M. D. Lima e R. Carmo. "Exact Algorithms for the Graph Coloring Problem". *Revista de Informática Teórica e Aplicada* 25.4 (2018), p. 57. DOI: 10.22456/2175-2745.80721.

[34] V. L. de Lima, M. Iori e F. K. Miyazawa. "Exact solution of network flow models with strong relaxations". *Mathematical Programming* 197 (2022), pp. 813–846. DOI: 10.1007/s10107-022-01785-9.

[35] E. Malaguti, M. Monaci e P. Toth. "An Exact Approach for the Vertex Coloring Problem". *Discrete Optimization* 8.2 (2011), pp. 174–190. ISSN: 1572-5286. DOI: 10.1016/j.disopt.2010.07.005.

[36] E. Malaguti e P. Toth. "A survey on vertex coloring problems". *International Transactions in Operational Research* 17.1 (2010), pp. 1–34. DOI: 10.1111/j.1475-3995.2009.00696.x.

[37] A. Mehrotra e M. A. Trick. "A Column Generation Approach for Graph Coloring". *INFORMS Journal on Computing* 8.4 (1996), pp. 344–354. ISSN: 1091-9856. DOI: 10.1287/ijoc.8.4.344.

[38] I. Méndez-Díaz e P. Zabala. "A Branch-and-Cut Algorithm for Graph Coloring". *Discrete Applied Mathematics* 154.5 (2006), pp. 826–847. ISSN: 0166-218X. DOI: 10.1016/j.dam.2005.05.022.

[39] I. Méndez-Díaz e P. Zabala. "A Cutting Plane Algorithm for Graph Coloring". *Discrete Applied Mathematics* 156.2 (2008), pp. 159–179. ISSN: 0166-218X. DOI: 10.1016/j.dam.2006.07.010.

[40] W. Meyer. "Equitable Coloring". *The American Mathematical Monthly* 80.8 (1973), pp. 920–922. DOI: 10.1080/00029890.1973.11993408.

[41] D. R. Morrison, E. C. Sewell e S. H. Jacobson. "Solving the Pricing Problem in a Branch-and-Price Algorithm for Graph Coloring Using Zero-Suppressed Binary Decision Diagrams". *INFORMS Journal on Computing* 28.1 (2016), pp. 67–82. ISSN: 1091-9856. DOI: 10.1287/ijoc.2015.0667.

[42] D. R. Morrison et al. "A Wide Branching Strategy for the Graph Coloring Problem". *INFORMS Journal on Computing* 26.4 (2014), pp. 704–717. DOI: 10.1287/IJOC.2014.0593.

[43] A. Pessoa, R. Sadykov e E. Uchoa. "Solving Bin Packing Problems Using VRPSolver Models". *Operations Research Forum* 2.20 (2021). DOI: 10.1007/s43069-020-00047-8.

[44] Schindl, David. "Some combinatorial optimization problems in graphs with applications in telecommunications and tomography". en (2005). DOI: 10.5075/EPFL-THESIS-3082.

[45] D. d. Werra. "An introduction to timetabling". *European Journal of Operational Research* 19 (2 1985), pp. 151–162. DOI: 10.1016/0377-2217(85)90167-5.