

---

---

Mestrado e Doutorado em Ciência da Computação (MDCC)  
Universidade Federal do Ceará (UFC)

---

---

# Um Estudo do Politopo e dos Limites Inferiores Gerados pela Formulação de Coloração dos Representantes

Victor Almeida Campos  
campos@lia.ufc.br

Agosto de 2005

**Banca Examinadora:**

- Ricardo Cordeiro Corrêa (MDCC/UFC)
- Manoel Bezerra Campêlo Neto (DC/UFC)
- Philippe Michelon (UECE)
- Cid Souza (IC - UNICAMP)



# Sumário

---

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Definições e Notações</b>	<b>4</b>
2.1	Teoria dos Grafos . . . . .	4
2.2	Problemas de Otimização . . . . .	11
2.3	Álgebra Linear . . . . .	12
2.4	Teoria Poliédrica . . . . .	13
2.5	Programação Linear . . . . .	14
2.6	Dualidade . . . . .	16
2.7	Programação Inteira . . . . .	17
<b>3</b>	<b>Formulações de Programação Inteira para o Problema de Coloração</b>	<b>19</b>
3.1	Parâmetros de Comparação . . . . .	20
3.2	Conjuntos Independentes . . . . .	21
3.2.1	Formulação . . . . .	21
3.2.2	Resolvendo o Problema Linear . . . . .	22
3.2.3	Discussão . . . . .	23
3.3	Cores por Vértice . . . . .	25

3.3.1	Formulação . . . . .	25
3.3.2	Facetas Baseadas em Subestruturas . . . . .	26
3.3.3	Discussão . . . . .	27
3.4	Orientações Acíclicas . . . . .	28
3.4.1	Formulação . . . . .	28
3.4.2	Discussão . . . . .	31
3.5	Tabela de Parâmetros das Formulações . . . . .	31
<b>4</b>	<b>Formulação dos Representantes</b>	<b>35</b>
4.1	Formulação . . . . .	35
4.2	Facetas e Politopo Associado . . . . .	39
4.3	Discussão . . . . .	44
<b>5</b>	<b>Representantes por Planos-de-Corte Faciais</b>	<b>47</b>
5.1	Pré-processamento . . . . .	48
5.1.1	Remoção de Vértices . . . . .	48
5.1.2	Decomposição por Componentes Conexas e Componentes Conexas do Complemento . . . . .	53
5.2	Tratando o Problema Linear . . . . .	54
5.2.1	Modelo Inicial . . . . .	55
5.2.2	Seleção de Inequações . . . . .	56
5.3	Heurísticas de Separação para Facetas Externas . . . . .	58
5.3.1	Cliques Externas . . . . .	58
5.3.2	Buracos Externos . . . . .	61
5.4	Algoritmo de Planos de Corte . . . . .	63
<b>6</b>	<b>Experimentos e Resultados</b>	<b>66</b>
6.1	Descrição de Instâncias . . . . .	66
6.2	Sumário de Implementação . . . . .	69

6.3	Sumário de Resultados . . . . .	70
<b>7</b>	<b>Conclusões</b>	<b>82</b>
<b>A</b>	<b>Demonstrações das Facetas dos Representantes</b>	<b>84</b>
A.1	Trecho do Artigo [7] com Demonstrações . . . . .	84

## Lista de Tabelas

---

3.1	Parâmetros comparativos sobre as formulações. . . . .	34
6.1	Sumário de instâncias de grafos aleatórios ou quase-aleatórios . . . . .	71
6.2	Sumário de instâncias de grafos de registros e outros grafos diversos . . . . .	72
6.3	Sumário de instâncias de grafos queen . . . . .	73
6.4	Sumário de instâncias de grafos mycielski e mycielski generalizados . . . . .	74
6.5	Sumário de resultados sobre grafos aleatórios ou quase-aleatórios . . . . .	75
6.6	Sumário de resultados sobre grafos de registros, queen e outros grafos diversos	76
6.7	Sumário de resultados sobre grafos mycielski e mycielski generalizados . . . . .	77
6.8	Sumário de cortes gerados sobre grafos aleatórios ou quase-aleatórios . . . . .	78
6.9	Sumário de cortes gerados sobre grafos de registros, queen e outros grafos diversos . . . . .	79
6.10	Sumário de cortes gerados sobre grafos mycielski e mycielski generalizados	80

## Lista de Figuras

---

2.1	Representação gráfica de grafos. Um vértice será representado por uma circunferência, uma aresta por uma ligação entre a representação de dois vértices e um arco por uma ligação com uma seta em uma das pontas. Apresentamos um grafo simples e um grafo direcionado nas figuras 2.1(a) e 2.1(b), respectivamente. . . . .	5
2.2	Exemplos de definições de adjacência, vértices dominados e vértices universais. . . . .	6
2.3	Os grafos $C_7$ , $\bar{C}_7$ e $W_5$ podem ser observados nas figuras 2.3(a), 2.3(b) e 2.3(c), respectivamente. . . . .	8
2.4	Uma coloração pode ser vista como uma atribuição de valores (2.4(a)) ou como uma família de classes de cor (2.4(b)). Um exemplo de multicoloração pode ser visto na Figura 2.4(c). Neste caso, foi utilizado a família de classes de cor para representar esta multicoloração. . . . .	10
2.5	Uma 5/2-coloração para o grafo $C_5$ . . . . .	11
3.1	Solução da relaxação de CIM aplicada ao $c_5$ . Os conjuntos independentes foram circulados com as linhas tracejadas e o valor das variáveis correspondentes são apresentados na figura. . . . .	24

3.2	Aplicação do operador $D$ . O grafo direcionado apresentado em 3.2(b) é o resultado da aplicação do operador $D$ sobre o grafo apresentado em 3.2(a)	28
3.3	Um grafo $D(G)$ gerado pelo operador $D$ pode ser observado na Figura 3.3(a). A Figura 3.3(b) mostra a construção do grafo $D'(G)$ . A Figura 3.3(c) exhibe um fluxo unitário de $s$ a $t$ que passa por $u, v, w$ e $z$ . O fluxo é representado pelos arcos escuros enquanto os arcos que não são percorridos pelo fluxo estão pontilhados. Este fluxo representa o caminho direcionado de $u$ para $z$ que passa por $v$ e $w$ .	30
3.4	Geração de uma orientação acíclica que define uma coloração $S_1, S_2, S_3$ . Na Figura 3.4(a), apresentamos uma coloração do grafo $C_5$ . Orientamos para fora de $S_1$ as arestas que incidem sobre os seus vértices na Figura 3.4(b).	32
3.5	Dois soluções viáveis simétricas, geradas a partir da mesma coloração. Estas soluções viáveis foram geradas a partir da permutação $(i_1, i_2, i_3)$ , como indicado na figura.	33
4.1	Definimos uma orientação e uma coloração para o grafo $G = C_5$ na Figura 4.1(a). Na Figura 4.1(b), exibimos a orientação de $\bar{G}$ . A coloração utilizando os conceitos de representantes é apresentada na Figura 4.1(c) e os conjuntos de vértices fonte e sumidouro na Figura 4.1(d).	37
4.2	Descrevemos uma coloração na Figura 4.2(a). A representação desta coloração utilizando as variáveis do modelo dos representantes é apresentada na Figura 4.2(b).	38
4.3	Ilustração dos conceitos de aresta segura e $G^{\text{safe}}$ .	40
4.4	Uma inequação interna violada com $H = \{v_1, \dots, v_5\} \subseteq \bar{N}^+(u)$ .	41
4.5	Esquema das variáveis envolvidas no somatório da inequação interna. As variáveis envolvidas são referentes aos vértices e arcos tracejados.	42



4.6	Uma solução ótima que define o valor de $C_5$ desta formulação. As arestas do grafo estão representadas por linhas escuras enquanto as variáveis referentes aos à orientação do complemento estão representadas pelas setas tracejadas. O valor indicado em um vértice $u$ indica o valor de $x(u, u)$ e o valor indicado em um arco $uv$ indica o valor de $x(u, v)$ . . . . .	45
5.1	Exemplo da decomposição de um grafo. . . . .	52
A.1	External inequality. . . . .	85
A.2	Some subgraphs related to external facets. . . . .	87
A.3	Example to show that inequalities (4.1)-(4.2) do not imply $\Phi(H_i) \geq 2$ , $i \in \{1, \dots, \ell\}$ , when integrality constraints are replaced by $0 \leq x_{uv} \leq 1$ , for all $u \in V \setminus T$ and $v \in \bar{N}^+[u]$ . Dashed lines indicate the variables whose values are shown in the figure. The values of the variables associated with the vertices are also shown. . . . .	91

# Introdução

---

Dado um inteiro positivo  $k$ , uma coloração ou  $k$ -coloração de  $G = (V, E)$  é uma atribuição de valores de  $\{1, \dots, k\}$  aos vértices de  $G$  tal que cada vértice receba pelo menos um valor e que vértices adjacentes recebam valores diferentes. O problema de coloração de vértices é definido como o problema de encontrar o menor valor  $\chi(G)$ , conhecido como número cromático de  $G$ , tal que  $G$  admite uma  $\chi(G)$ -coloração.

Este problema é um dos modelos mais estudados em Teoria dos Grafos pela sua relevância em campos práticos e teóricos. Do ponto de vista teórico, o problema de coloração é NP-Difícil [23, 31]. Além disto, foi classificado entre os problemas mais difíceis de NP, no sentido de que achar uma aproximação para o número cromático é um problema computacionalmente difícil [36]. Do ponto de vista prático, os algoritmos exatos existentes atualmente resolvem de forma eficiente apenas instâncias com poucas centenas. Devido a esta limitação, instâncias grandes costumam ser resolvidas por métodos heurísticos [26, 28]. Também é possível usar este problema para modelar uma coleção de problemas de interesse prático como escalonamento [16], alocação de frequências [22], alocação de registros [11, 12] e método de elementos finitos [41].

A importância do problema de coloração tem incentivado a investigação de métodos para encontrar limites inferiores próximos do número cromático [30]. Historicamente, as primeiras tentativas para resolvê-lo lidavam com cliques maximais. O primeiro algoritmo

exato para resolver este problema foi apresentado por Brown [5]. Este algoritmo consiste de uma enumeração implícita de todas as colorações de um grafo. Em seguida, Brelaz [4] apresentou o algoritmo *DSATUR*. Este algoritmo é baseado no método de *branch-and-bound*, utilizando o valor de uma clique de provável valor máximo para limitar o número de subproblemas gerados. Outros algoritmos surgiram com melhorias para o método *DSATUR* [24, 34, 40, 42], mas sempre utilizando o valor de uma clique de provável valor máximo como limitante inferior.

Para melhorar o limite inferior, Mehrotra e Trick utilizaram a formulação de conjuntos independentes, chamada de CIM, que tem uma variável binária para cada conjunto independente maximal  $W$  [37]. Esta formulação obtém o valor  $\chi_F(G)$ , o número cromático fracionário, como valor de sua relaxação linear. Nesse trabalho, Mehrotra e Trick mostraram a força do limite inferior fornecido por  $\chi_F(G)$  mesmo sendo necessária a solução de problemas computacionalmente difíceis. Como o método de geração de colunas proposto requer a solução do problema de conjunto independente máximo ponderado, um problema computacionalmente difícil, técnicas padrão para reforçar integralidade são de difícil aplicação para gerar conjuntos independentes melhores. Isto desencoraja o uso de planos de corte para melhorar os limites inferiores obtidos com esta formulação.

Outras alternativas para gerar limites inferiores surgiram na literatura na forma de formulações de programação inteira 0-1. Por exemplo, Díaz e Zabala trabalharam no estudo do politopo da formulação clássica [18], chamada de CPV, Figueiredo, Barbosa, Maculan e Souza numa formulação proposta [20], baseada nas orientações acíclicas do grafo, chamada de OA, e Campêlo, Corrêa e Frota também propuseram uma formulação [9], baseada em vértices representantes de cores, chamada de REP. Estas poderiam ser utilizadas para gerar limites inferiores aprimorados por não possuírem um número exponencial de variáveis e pela existência dos estudos parciais dos politopos associados.

A formulação REP consiste em escolher vértices para serem os representantes das classes de cor. Em um trabalho posterior, a formulação foi revisada para remover a simetria existente [8]. Neste trabalho, a formulação REP é comparada com a formulação

de conjuntos independentes. Este trabalho fornece formas para atacar alguns desafios lançados pela implementação de Mehrotra e Trick. Com um subconjunto das facetas descritas, é possível resolver a relaxação linear da formulação REP para obter um limite inferior para o número cromático fracionário. Utilizando as demais facetas, podemos, até, obter um limite inferior que supere  $\chi_F(G)$ .

Este trabalho tem alguns objetivos relacionados a formulações de programação inteira 0-1 para o problema de coloração. O primeiro objetivo consiste em comparar as formulações conhecidas. O segundo consiste em fazer um estudo mais aprofundado do politopo da formulação de coloração dos representantes, considerando remoções de simetria não tratadas no trabalho inicial. Enfim, queremos utilizar o método de planos de corte [15] para fornecer limites inferiores fortes em relação ao número cromático usando a formulação dos representantes. Uma comparação entre as formulações é usada para indicar a escolha pela formulação REP. Dada esta escolha, utilizamos esta formulação para gerar limites inferiores para  $\chi_F(G)$ . Observamos que é possível obter valores muito próximos de  $\chi_F(G)$  utilizando uma fração do tempo necessário para resolver a formulação CIM. Experimentos computacionais foram realizados para comparar o limite inferior fornecido com o número cromático fracionário.

Os capítulos restantes estão organizados da seguinte maneira. No Capítulo 2 veremos as definições e notações utilizados no resto do texto. O Capítulo 3 definirá os termos a serem usados na comparação entre as formulações e estes termos serão analisados para as formulações CIM, CPV e OA. O Capítulo 4 trará a avaliação da formulação REP e um estudo parcial do politopo inteiro associado. No Capítulo 5 veremos detalhes da implementação e os resultados de comparação serão apresentados no Capítulo 6. O Capítulo 7 contém algumas conclusões e direções para trabalhos futuros.

## Definições e Notações

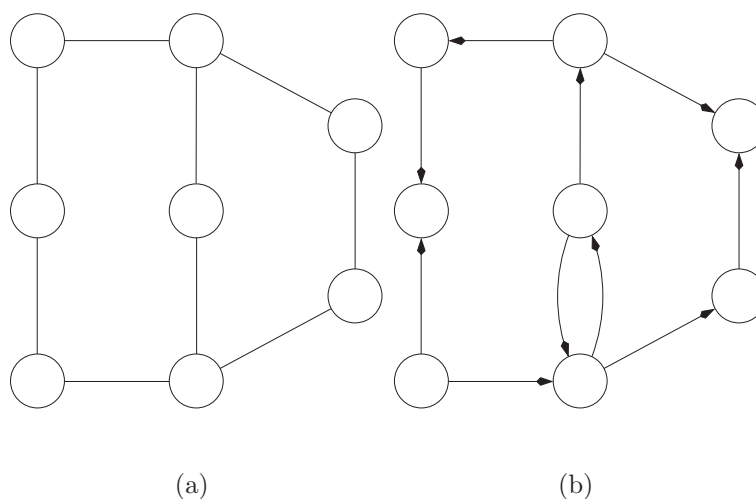
---

A notação utilizada ao longo do texto é consistente com a terminologia geralmente aceita em Teoria dos Grafos, Álgebra Linear, Programação Linear e Programação Inteira. A intenção deste capítulo é de definir notações e terminologias que serão utilizados durante a dissertação. Esta notação pode ser encontrada em vários outros trabalhos [2, 19, 21, 23, 39]. Na Seção 2.1 são apresentados os conceitos fundamentais em grafos assim como a definição de alguns problemas clássicos da Teoria dos Grafos. Na Seção 2.3 revisaremos alguns conceitos importantes de Álgebra Linear. Em seguida, na Seção 2.2 formalizaremos a noção de um problema de otimização. Traremos alguns conceitos básicos necessários sobre Teoria Poliédrica na Seção 2.4. Questões de Programação Linear serão tratadas na Seção 2.5, enquanto na Seção 2.6 trataremos o relacionamento com o seu dual. Finalmente, na Seção 2.7 será abordado um problema de programação inteira, assim como será discutido o método de planos-de-corte.

### 2.1. Teoria dos Grafos

Um *grafo*  $G = (V, E)$  é um par ordenado de conjuntos, onde  $E \subseteq V^2$ . Os elementos do conjunto  $V$  são chamados de *vértices*, enquanto os elementos de  $E$  são chamados de *arestas*. Utilizamos  $V(G)$  e  $E(G)$ , respectivamente, para o *conjunto de vértices* e o

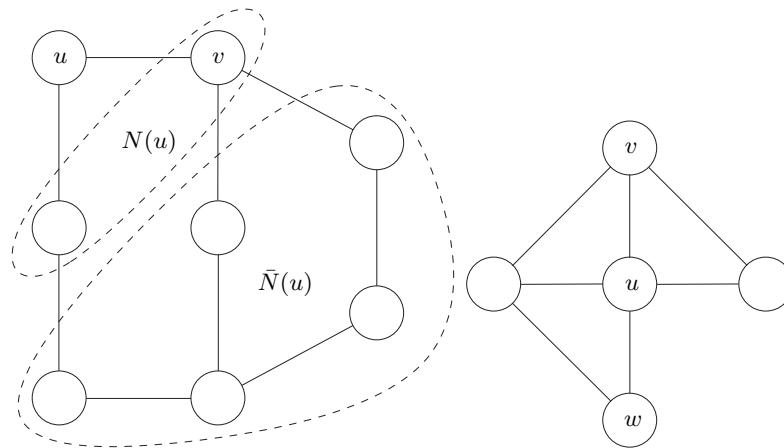
conjunto de arestas de  $G$ . Em um grafo *direcionado*, consideramos que, se  $u \in V$  e  $v \in V$ , então  $(u, v) \neq (v, u)$ . Neste caso, um elemento de  $E$  será chamado de um *arco*. Em um grafo *não direcionado*, consideramos os elementos de  $E$  como pares não ordenados. Podemos utilizar a notação  $uv$  para nos referir a um par  $(u, v) \in V^2$  em grafos não direcionados e  $\overrightarrow{uv}$  para representar arcos em grafos direcionados. Para um grafo não direcionado  $G$ , o *complemento* de  $G$  é  $\bar{G} = (V, \bar{E})$ , tal que, para todo  $uv \in V^2$ ,  $uv \in \bar{E}$  se, e somente se,  $uv \notin E$  e  $u \neq v$ . Denotamos por  $n$  a cardinalidade do conjunto de vértices  $V$ , por  $m$  a cardinalidade do conjunto de arestas  $E$  e por  $\bar{m}$  a cardinalidade de  $\bar{E}$ . Chamaremos de *laço* um par  $vv \in E$ . Um grafo é dito *simples* se for não direcionado e não possuir *laços*. A forma como podemos representar um grafo graficamente pode ser encontrada na Figura 2.1.



**Figura 2.1.** Representação gráfica de grafos. Um vértice será representado por uma circunferência, uma aresta por uma ligação entre a representação de dois vértices e um arco por uma ligação com uma seta em uma das pontas. Apresentamos um grafo simples e um grafo direcionado nas figuras 2.1(a) e 2.1(b), respectivamente.

Dois vértices  $u, v \in V$  são *adjacentes* em  $G$  se  $uv \in E$ . A *vizinhança* de  $u$  é definida por  $N_G(u) = \{v \in G \mid uv \in E\}$  e a sua *anti-vizinhança* por  $\bar{N}_G(u) = V \setminus (N_G(u) \cup \{u\})$ . Em

ambos os casos,  $G$  pode ser omitido quando o grafo estiver claro pelo contexto. O *grau* de um vértice  $u$ , denotado por  $d(u)$ , é a cardinalidade de  $N(u)$ . Utilizamos  $N[u] = N(u) \cup \{u\}$  e  $\bar{N}[u] = \bar{N}(u) \cup \{u\}$ . Exemplos dos conceitos de adjacência e de vizinhança podem ser encontrados na Figura 2.2(a). Um vértice  $u$  é dito *universal* se  $\bar{N}(u) = \emptyset$ . Um grafo  $G$  é dito *completo* se todos os seus vértices forem universais. Um vértice  $u$  é dito *dominado por*  $v$  se  $N(u) \subseteq N(v)$ . Se existe um vértice  $v$  que domina  $u$ ,  $u$  será dito um *vértice dominado*.



(a) Neste grafo,  $u$  é adjacente a  $v$ .  
Exibimos os conjuntos de vizinhos  
e anti-vizinhos de  $u$ .

(b) Neste grafo,  $u$  é  
um vértice universal e  
o vértice  $w$  é dominado  
por pelo vértice  $v$ .

**Figura 2.2.** Exemplos de definições de adjacência, vértices dominados e vértices universais.

Um grafo  $G'$  é um *subgrafo* de  $G$  se  $V(G') \subseteq V(G)$  e  $E(G') \subseteq E(G)$ , o que pode ser denotado por  $G' \subseteq G$ . Um subgrafo  $G' \subseteq G$  é um *subgrafo induzido* se, para todo  $uv \in E(G)$ ,  $uv$  pertence a  $E(G')$  sempre que  $u, v \in V(G')$ . O subgrafo induzido por  $V' \subseteq V(G)$  é definido por  $G[V'] = (V', E(G) \cap V' \times V')$ . Um subgrafo  $G' \subseteq G$  é dito um *subgrafo próprio* de  $G$  se  $G' \neq G$  e é denotado por  $G' \subset G$ . Uma *clique*  $K$  de  $G$

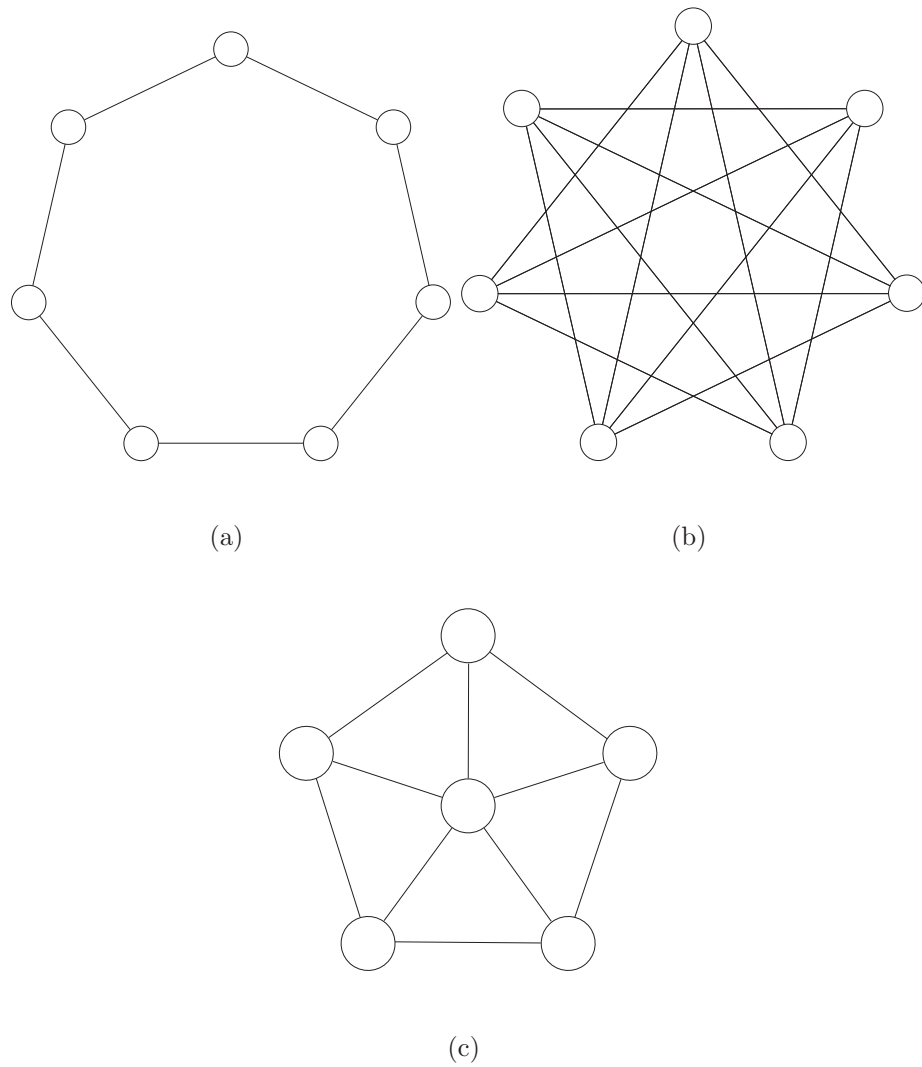
é um subconjunto de  $V(G)$  tal que  $G[K] = (K, K \times K \setminus \{vv \mid v \in K\})$ . O *tamanho máximo de uma clique* de  $G$  é representado por  $\omega(G)$ . Uma clique  $K$  é dita *maximal* se não existir outra clique de  $G$  que contenha  $K$ . Um *conjunto independente*  $W \subseteq G$  é um subconjunto de  $V(G)$  tal que  $G[W] = (W, \emptyset)$  e é chamado de *maximal* se não é contido em outro conjunto independente de  $G$ .

Um conjunto de arestas  $P = \{v_0v_1, v_1v_2, \dots, v_{k-1}v_k\}$ , com  $v_i \neq v_j$  para  $i \neq j$ , será chamado de *caminho*. Dizemos que  $P$  é um caminho de  $v_0$  a  $v_k$  e tem *comprimento*  $k$ . A *distância* entre dois vértices  $u$  e  $v$  é definida como o menor comprimento de um caminho entre  $u$  e  $v$ . Caso tal caminho não exista, dizemos que a distância é infinita. Caso  $v_0$  seja adjacente a  $v_k$  em um caminho  $P$ , com comprimento maior ou igual a 2, chamamos o conjunto  $C = P \cup \{v_0v_k\}$  um *ciclo* de tamanho  $k + 1$ . Um grafo é *conexo* se possui um caminho entre quaisquer dois de seus vértices. Caso contrário, é dito *desconexo*. Denominamos por *componente* um subgrafo conexo maximal.

Chamamos um subgrafo  $G' \subseteq G$  de *buraco* se este é um ciclo induzido em  $G$ . Um buraco será dito um *buraco ímpar* se ele tem uma quantidade ímpar de vértices. Um buraco com  $k$  vértices é denotado por  $C_k$ . Um subgrafo  $G' \subseteq G$  será dito um *anti-buraco* se  $G'$  é o complemento de um ciclo induzido em  $G$ . Analogamente, podemos definir um *anti-buraco ímpar* como sendo um anti-buraco com uma quantidade ímpar de vértices. Um anti-buraco com  $k$  vértices é denotado por  $\bar{C}_k$ . Chamamos um subgrafo  $G' \subseteq G$  de *roda com centro* se existe um vértice  $u \in V(G')$ , universal em  $G'$ , tal que  $G'[V(G') \setminus \{u\}]$  define um buraco. Denotamos uma roda com  $k + 1$  vértices por  $W_k$ . Exemplos destas estruturas podem ser observados na Figura 2.3.

Uma *orientação* de  $G$  é uma função  $\sigma : E \rightarrow V$  tal que  $\sigma(uv) \in \{u, v\}$ . O grafo direcionado obtido a partir de  $G$  com a orientação  $\sigma$  é denotado por  $G_\sigma$ . Defina a *vizinhança positiva* de  $u$  por  $N^+(u) = \{v \in N(u) \mid \sigma(uv) = v\}$  e a *vizinhança negativa* por  $N^-(u) = \{v \in N(u) \mid \sigma(uv) = u\}$ . Se uma orientação é definida para  $\bar{G}$ , as *anti-vizinhanças positiva* e *negativa* podem ser definidas de forma similar e são denotados por  $\bar{N}^+(u)$  e  $\bar{N}^-(u)$ , respectivamente. Esta notação pode ser estendida de  $(u)$



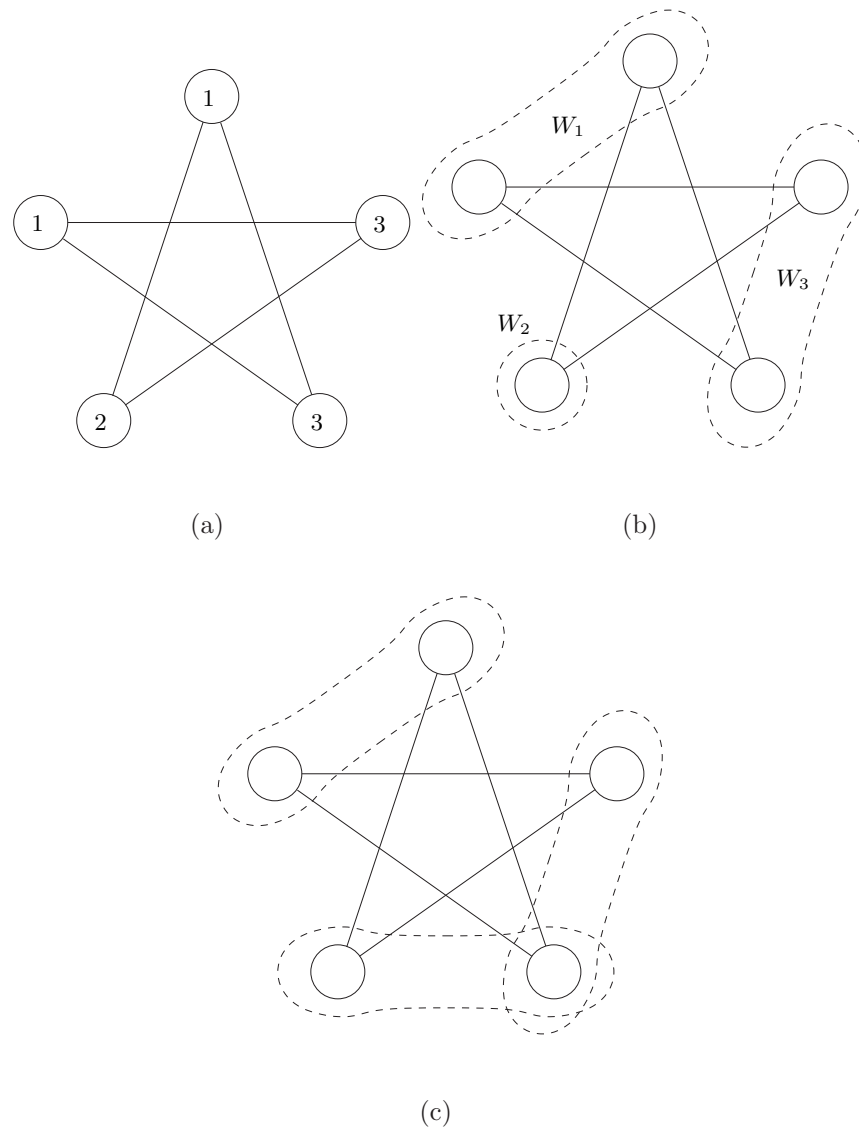


**Figura 2.3.** Os grafos  $C_7$ ,  $\bar{C}_7$  e  $W_5$  podem ser observados nas figuras 2.3(a), 2.3(b) e 2.3(c), respectivamente.

para  $[u]$  ao incluir  $u$  nos conjuntos. Um *caminho direcionado* é um conjunto de arcos  $\vec{p} = \{\overrightarrow{u_0u_1}, \overrightarrow{u_1u_2}, \dots, \overrightarrow{u_ku_{k+1}}\}$  tal que, para  $i \neq j$ , temos que  $u_i \neq u_j$ . O tamanho de um caminho direcionado equivale à cardinalidade de  $\vec{p}$  e pode ser denotado por  $|\vec{p}|$ . Um *ciclo direcionado* é um conjunto de arcos  $\vec{C} = \{\overrightarrow{u_0u_1}, \overrightarrow{u_1u_2}, \dots, \overrightarrow{u_{k-1}u_k}, \overrightarrow{u_ku_0}\}$  tal que  $\vec{C} \setminus \{\overrightarrow{u_ku_0}\}$  define um caminho direcionado. Uma orientação de  $G$  é dita acíclica se não possui ciclos direcionados.

Dado um inteiro positivo  $k$ , uma *coloração* ou  $k$ -*coloração* de  $G$  é uma atribuição de valores do conjunto  $\{1, \dots, k\}$  aos vértices de  $G$  tal que cada vértice recebe um valor e as extremidades de uma aresta recebem valores diferentes. Uma coloração de  $G$  também pode ser vista como uma família  $W_1, \dots, W_k$  de  $k \geq \chi(G)$  conjuntos independentes de  $G$ , cada conjunto independente  $W_i$  na família definindo uma classe de cor associada à cor  $i$ . A Figura 2.4 pode ser usada para visualizar estas duas definições. O problema de coloração de grafos é definido como o problema de encontrar o menor número de cores  $\chi(G)$ , conhecido como *número cromático*, tal que  $G$  possui uma  $\chi(G)$ -coloração. Os valores do conjunto  $\{1, \dots, k\}$  também podem ser chamados de cores quando associados ao problema de coloração e o conjunto de vértices com a mesma cor é chamado de *classe de cor*. Caso seja atribuída mais de uma cor para algum vértice, também podemos nos referir a esta coloração como uma *multicoloração*. Um grafo  $G$  é dito *crítico* ou  $\chi(G)$ -*crítico* se, para todo subgrafo induzido próprio  $G' \subset G$ ,  $\chi(G') < \chi(G)$ .

Dados inteiros positivos  $k$  e  $\ell$ , uma  $k/\ell$ -*coloração* de  $G$  é uma atribuição  $c : V \rightarrow 2^{\{1, \dots, k\}}$  com  $|c(v)| = \ell$ , para todo  $v \in V$ , de tal forma que os subconjuntos atribuídos a vértices adjacentes sejam disjuntos. O *número cromático fracionário*, representado por  $\chi_F(G)$ , é a menor razão  $k/\ell$  tal que  $G$  possui uma  $k/\ell$ -coloração [3, 25, 27, 35]. Uma  $5/2$ -coloração para o grafo  $C_5$  pode ser vista na Figura 2.5. É simples observar que  $\omega(G) \leq \chi_F(G) \leq \chi(G)$ . O limitante superior pode ser obtido ao observar que uma  $\chi(G)$ -coloração de  $G$  pode ser utilizada para construir uma  $k/1$ -coloração. O limitante inferior pode ser obtido ao observar que, dada uma clique de tamanho  $\omega(G)$ , os conjuntos atribuídos aos vértices desta clique devem ser disjuntos, implicando que  $k \geq \ell\omega(G)$ .



**Figura 2.4.** Uma coloração pode ser vista como uma atribuição de valores (2.4(a)) ou como uma família de classes de cor (2.4(b)). Um exemplo de multicoloração pode ser visto na Figura 2.4(c). Neste caso, foi utilizado a família de classes de cor para representar esta multicoloração.

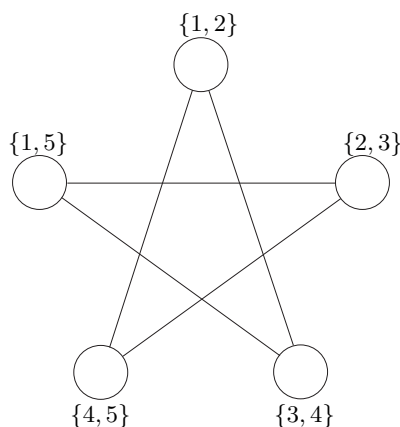


Figura 2.5. Uma  $5/2$ -coloração para o grafo  $C_5$ .

## 2.2. Problemas de Otimização

Um *problema de otimização*  $\Pi$  é um par  $(X, f)$ , onde  $X$  é um conjunto, chamado *conjunto de soluções viáveis* de  $\Pi$ , e  $f : X \rightarrow \mathbb{R}$  é a sua *função de custo*. O objetivo de um problema de otimização  $\Pi$  é encontrar uma solução viável que minimize ou maximize a função de custo. No caso de  $\Pi$  ser um problema de minimização, deseja-se encontrar um ponto  $x^* \in X$  tal que  $f(x^*) \leq f(x)$ , para qualquer  $x \in X$ . Por outro lado, no caso de um problema de maximização, o objetivo é encontrar  $x^* \in X$  tal que  $f(x^*) \geq f(x)$ , para qualquer  $x \in X$ . Um ponto  $x^*$  com tais propriedades é chamado de *solução ótima*.

Um exemplo de problema de otimização é o problema de coloração de vértices, definido na Seção 2.1. Neste problema, o conjunto de soluções viáveis inclui as colorações de  $G$  enquanto a função de custo representa o número de cores distintas utilizadas em uma coloração.

Uma forma genérica de resolver um problema de otimização  $(X, f)$  é utilizando o *método branch-and-bound*. A idéia do método *branch-and-bound* consiste em enumerar todas as soluções viáveis de um problema em uma estrutura de árvore de decisão em que a cada nó desta árvore está associado um subproblema de otimização. Esta árvore é construída de forma que a sua raiz está associada a  $(X, f)$ . Se  $v$  é um nó da árvore

associado ao problema  $(M, f)$ , então a cada um de seus filhos  $u_i$  está associado o problema  $(N_i, f)$  de forma que os conjuntos  $N_i$  sejam uma partição de  $M$ .

A busca por uma solução ótima utilizando uma árvore *branch-and-bound* pode ser mais eficiente se conhecermos boas heurísticas de geração de soluções viáveis e bons limitantes inferiores (considerando um problema de minimização). Se em um dado nó desta árvore o valor do limitante inferior para este nó for superior ao valor de alguma solução viável conhecida, então a solução ótima do problema não pode estar em um descendente deste nó, e podemos considerar toda a subárvore a partir deste nó como visitada. Deve ser observado que a busca por melhores limitantes inferiores reduz drasticamente o tempo necessário para calcular uma solução ótima por métodos baseados em *branch-and-bound*. Tal busca por melhores limitantes inferiores pode ser observado na literatura para o problema de coloração em [4, 5, 24, 30, 34, 40, 42].

### 2.3. Álgebra Linear

Denotamos por  $\mathbb{R}$ ,  $\mathbb{Z}$  e  $\mathbb{N}$  os conjuntos de números reais, inteiros e naturais, respectivamente. Para  $n \in \mathbb{N}$ , o símbolo  $\mathbb{R}^n$  denota o conjunto de todos os vetores com  $n$  componentes reais. A menos que se especifique o contrário, vetores serão vetores-coluna. Denotamos por  $\mathbb{R}^{m \times n}$  o conjunto das matrizes reais com  $m$  linhas e  $n$  colunas. Quando utilizado como vetor, denominaremos  $0 \in \mathbb{R}^n$  e  $1 \in \mathbb{R}^n$  para representar os vetores onde todos os elementos são 0's ou 1's, respectivamente. Seja  $A \in \mathbb{R}^{m \times n}$  uma matriz, um elemento na  $i$ -ésima linha e  $j$ -ésima coluna é denotado por  $A_{ij}$ . A  $j$ -ésima coluna de  $A$  é um vetor com  $m$  elementos, denotado por  $A_{*j}$ . Analogamente, a  $i$ -ésima linha de  $A$  é denotada por  $A_{i*}$ . Dado um vetor  $a \in \mathbb{R}^n$ , o  $j$ -ésimo elemento deste vetor é representado por  $a_j$ .

Um vetor  $x \in \mathbb{R}^n$  é uma *combinação linear* dos vetores  $x_1, \dots, x_k \in \mathbb{R}^n$  se, para algum  $\alpha \in \mathbb{R}^k$ ,

$$x = \sum_{i=1}^k \alpha_i x_i.$$

Uma tal combinação linear é chamada de *afim* se  $1^T \alpha = 1$ , é dita *cônica* se  $\alpha \geq 0$  e *convexa* se for afim e cônica. Para um conjunto não-vazio  $S \subseteq \mathbb{R}^n$ , denominamos o *fecho linear* de  $S$  como sendo o conjunto de todos os vetores que são combinação linear de um número finito de vetores de  $S$ . Analogamente, definimos os *fechos afim*, *cônico* e *convexo*.

Um conjunto  $S \subseteq \mathbb{R}^n$  é dito *linearmente independente* se, para qualquer subconjunto finito  $\{x_1, \dots, x_k\} \subseteq S$ , sempre que  $\sum_{i=1}^k \alpha_i x_i = 0$ , com  $\alpha \in \mathbb{R}^k$ , temos que  $\alpha = 0$ . Dizemos que  $S$  é *afim-independente* se, para qualquer subconjunto finito  $\{x_1, \dots, x_k\} \subseteq S$ , a igualdade  $\sum_{i=1}^k \alpha_i x_i = 0$ , com  $\alpha \in \mathbb{R}^k$  e  $1^T \alpha = 0$ , implica que  $\alpha = 0$ . Um conjunto  $S$  é *linearmente dependente* se ele não é linearmente independente. Definimos *afim-dependente* de forma análoga.

Para  $S \subseteq \mathbb{R}^n$ , o *posto* de  $S$ , denotado por  $\text{posto}(S)$ , é a cardinalidade de um maior subconjunto de  $S$  que é linearmente independente. Analogamente, definimos o *posto afim* de  $S$ , denotado por  $\text{postoafim}(S)$ , como sendo a cardinalidade de um maior conjunto afim-independente contido em  $S$ . Prova-se que para todo  $S \subseteq \mathbb{R}^n$ ,  $\text{postoafim}(S) = \text{posto}(S) + 1$  se 0 pertence ao fecho afim de  $S$  e  $\text{postoafim}(S) = \text{posto}(S)$  caso contrário. O posto de uma matriz  $A$ , denotado por  $\text{posto}(A)$ , é o posto do conjunto de vetores-coluna de  $A$ , que prova-se ser igual ao posto do conjunto de vetores-linha de  $A$ . Uma matriz  $A \in \mathbb{R}^{m \times n}$  tem *posto-linha completo* se  $\text{posto}(A) = m$  e *posto-coluna completo* se  $\text{posto}(A) = n$ . Para  $S \subseteq \mathbb{R}^n$ , a *dimensão*  $\dim(S)$  de  $S$  é definida como  $\dim(S) = \text{postoafim}(S) - 1$ . Dizemos que  $S \subseteq \mathbb{R}^n$  tem *dimensão plena* se  $\dim(S) = n$ .

## 2.4. Teoria Poliédrica

Um subconjunto  $P \subseteq \mathbb{R}^n$  é chamado de *poliedro* se  $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$  para alguma matriz  $A \in \mathbb{R}^{m \times n}$  e algum vetor  $b \in \mathbb{R}^m$ . Um poliedro  $P$  definido por uma matriz  $A$  e um vetor  $b$  é denotado por  $P(A, b)$ . Se  $a \in \mathbb{R}^n \setminus \{0\}$  e  $\alpha \in \mathbb{R}$ , então os poliedros  $\{x \in \mathbb{R}^n \mid a^T x \leq \alpha\}$  e  $\{x \in \mathbb{R}^n \mid a^T x = \alpha\}$  são chamados de *semi-espaco* e de *hiperplano*, respectivamente.

Sejam  $P \subseteq \mathbb{R}^n$ ,  $a \in \mathbb{R}^n$  e  $\alpha \in \mathbb{R}$ . A inequação  $a^T x \leq \alpha$  é *válida* em relação a  $P$

se  $P \subseteq \{x \in \mathbb{R}^n \mid a^T x \leq \alpha\}$ . Quando o conjunto  $P$  está claro pelo contexto, dizemos simplesmente que a inequação é válida. Um conjunto  $F \subseteq P$  é uma *face* de  $P$  se existe uma inequação  $a^T x \leq \alpha$ , válida em relação a  $P$ , tal que  $F = P \cap \{x \in \mathbb{R}^n \mid a^T x = \alpha\}$ . Dizemos que  $F$  é uma face *própria* se  $F \neq P$  e *não-trivial* se  $\emptyset \neq F \neq P$ . Uma face não-trivial  $F$  de um poliedro  $P$  é uma *faceta* se  $F$  não está contida em nenhuma outra face própria de  $P$ . Podemos encontrar condições necessárias e suficientes para descrever facetas. No caso de poliedros de dimensão plena, podemos obter o seguinte teorema, cuja demonstração pode ser encontrada em [19].

**Teorema 1.** *Seja  $P \subseteq \mathbb{R}^n$  um poliedro de dimensão plena e  $F = \{x \in P \mid c^T x = \gamma\}$  uma face de  $P$ . Então, os resultados a seguir são equivalentes:*

1.  $F$  é uma faceta de  $P$ ;
2.  $\dim(F) = n - 1$ ;
3. se  $d^T x \leq \delta$ ,  $d \neq 0$ , é uma inequação válida tal que  $F \subseteq \{x \in P \mid d^T x = \delta\}$ , então existe  $\alpha \in \mathbb{R}$ ,  $\alpha > 0$ , tal que  $d = \alpha c$  e  $\delta = \alpha \gamma$ .

## 2.5. Programação Linear

Uma função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  é uma *função linear* se, e somente se,  $f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$ , para todos  $x, y \in \mathbb{R}^n$  e  $\alpha, \beta \in \mathbb{R}$ . Para qualquer função linear  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  e qualquer número real  $b$ , a equação  $f(x) = b$  e as inequações  $f(x) \leq b$  e  $f(x) \geq b$  são lineares. Uma função linear  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  pode ser escrita na forma  $f(x) = c^T x$ , com  $c \in \mathbb{R}^n$ .

Um *problema de programação linear* (PPL) é um problema de otimização  $(P, f)$  onde  $P$  é um poliedro e  $f$  é uma função linear. Um PPL pode ser expresso sob a *forma padrão*:

$$\begin{array}{ll}
 \text{(PPL) Minimizar} & c^T x \\
 \text{Sujeito a} & Ax = b \\
 & x \geq 0
 \end{array}$$

onde  $c \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{m \times n}$  e  $b \in \mathbb{R}^m$ . Vamos representar este PPL por  $(P(A, b), c)$  e supor que  $A$  tenha posto-linha completo. O vetor  $c$  é denominado *vetor de coeficientes de custo* e o vetor  $x$  é o *vetor de variáveis* a serem determinadas pelo problema. As equações do tipo  $Ax = b$  são chamadas *restrições do problema*. As inequações do tipo  $x \geq 0$  são denominadas de *restrições de não negatividade*. A matriz  $A$  é denominada *matriz de restrições*.

Uma submatriz  $B \in \mathbb{R}^{m \times m}$  é denominada de *matriz base*, ou simplesmente *base*, de  $A$  se  $B$  é invertível. A matriz  $N$  composta pelas colunas de  $A$  que não pertencem a  $B$  é denominada de *matriz não básica*. Chamamos de *variáveis básicas* os termos do vetor  $x$  associados às colunas de  $A$  que pertencem a  $B$ , formando o vetor  $x_B$ . Analogamente, denotamos por  $x_N$  o subvetor de *variáveis não básicas* formado pelos termos do vetor  $x$  referentes às colunas de  $A$  que pertencem a  $N$ . Dada uma base  $B$  de  $A$ , dizemos que a solução  $x$ , onde  $x_B = B^{-1}b$  e  $x_N = 0$ , é uma *solução básica*. Se  $x_B \geq 0$ , então  $x$  é uma *solução básica viável*. O Teorema 2, cuja demonstração pode ser encontrada em [2], reduz um PPL ao problema de obter a melhor solução básica viável.

**Teorema 2.** *Se o PPL  $(P(A, b), c)$  tem solução ótima, então existe uma solução ótima do PPL  $(P(A, b), c)$  que é solução básica viável.*

Seja  $x_B = B^{-1}b$  e  $x_N = 0$  a solução básica do PPL  $(P(A, b), c)$  associada à base  $B$ . Sejam  $c_B$  e  $c_N$  os coeficientes do vetor  $c$  relativos às variáveis básicas e não básicas, respectivamente. O valor da função objetivo em  $x$  é dado por  $c^T x = c_B^T B^{-1}b$ . O vetor de multiplicadores duais relativos a  $x$  é  $\pi^T = c_B^T B^{-1}$ . O vetor de custos reduzidos associados à solução  $x$  é dado por  $c_N^T - c_B^T B^{-1}N = c_N^T - \pi^T N$ . Caso uma base  $B$  esteja associada a uma solução básica viável, dizemos que esta solução é *básica primal viável*. Caso o vetor de custos reduzidos seja não negativo, ou seja  $c_N^T - \pi^T N \geq 0$ , dizemos que a solução  $x$ , associada à base  $B$ , é *básica dual viável*. Estes dois conceitos podem ser associados para descrever uma solução ótima, como descrito no Teorema 3. Uma demonstração para o Teorema 3 pode ser encontrada em [2].



**Teorema 3.** *Se uma solução básica  $x$  é primal viável e dual viável associada a um PPL  $(P(A, b), c)$ , então  $x$  é solução ótima.*

Uma forma de resolver um PPL  $(P(A, b), c)$  consiste em utilizar o algoritmo *primal simplex*. Este método começa a partir de uma solução básica primal viável  $x$  associada a uma base  $B$ . Caso  $B$  não seja dual viável, o algoritmo iterativamente constrói novas bases, mantendo a viabilidade primal, até que a viabilidade dual seja obtida. Uma segunda forma de resolver um PPL  $(P(A, b), c)$  consiste em utilizar o algoritmo *dual simplex*. Este algoritmo funciona de forma semelhante ao algoritmo primal simplex, com a diferença de que as bases iteradas são dual viáveis e o algoritmo visa obter otimalidade ao atingir a viabilidade primal. A escolha por um destes algoritmos normalmente está relacionada a uma solução básica inicial conhecida ser primal ou dual viável. Mais detalhes sobre estes algoritmos podem ser encontrados em [2].

## 2.6. Dualidade

Associado a um problema de programação linear está um outro problema de programação linear denominado de *problema dual*. O problema dual do *problema de programação linear primal*  $(P(A, b), c)$  é definido por:

$$\begin{aligned} \text{(D) Maximizar} \quad & b^T \pi \\ \text{Sujeito a} \quad & A^T \pi \leq c \end{aligned}$$

Algumas relações importantes sobre os problemas primal e o dual podem ser vistas nos resultados a seguir:

**Proposição 4.** *Se  $x$  é uma solução viável para um PPL e  $\pi$  é uma solução viável para o seu dual  $D$ , então  $c^T x \geq b^T \pi$ .*

**Teorema 5.** *Sejam  $P$  e  $D$  os problemas primal e dual como enunciados anteriormente. Então, ambos os problemas têm solução ótima e os valores ótimos são iguais se, e somente se, esses problemas têm soluções viáveis.*

Para melhor enunciar um resultado importante que utilizamos neste texto, devemos observar que um mesmo PPL pode ser escrito de formas diferentes. Dentre elas, utilizamos a *forma canônica* que tem o seguinte formato:

$$\begin{aligned} \text{(PC) Minimizar} \quad & c^T x \\ \text{Sujeito a} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

onde  $c \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{m \times n}$  e  $b \in \mathbb{R}^m$ . O problema dual associado ao problema primal em forma canônica é:

$$\begin{aligned} \text{(DC) Maximizar} \quad & b^T \pi \\ \text{Sujeito a} \quad & A^T \pi \leq c \\ & \pi \geq 0 \end{aligned}$$

O Teorema 6 é conhecido como o Teorema de Folgas Complementares.

**Teorema 6.** *Sejam  $x^*$  e  $\pi^*$  soluções viáveis para o problema primal e dual em suas formas canônicas. Elas são respectivamente ótimos se, e somente se,*

$$(c_j - A_{*j}^T \pi^*) x_j^* = 0 \quad j \in \{1, \dots, n\}$$

e

$$\pi_i^* (A_{i*}^T x^* - b_i) = 0 \quad i \in \{1, \dots, m\}.$$

As demonstrações para os resultados apresentados nesta seção podem ser encontrados em [19, 2].

## 2.7. Programação Inteira

Um *problema de programação inteira* (PPI) é um problema de programação linear em que as variáveis devem ser números inteiros. Em termos matemáticos, um PPI pode ser

representado por

$$\begin{aligned}
 \text{(PPI) Minimizar} \quad & c^T x \\
 \text{Sujeito a} \quad & Ax = b \\
 & x \geq 0 \\
 & x \in \mathbb{Z}^n
 \end{aligned}$$

Representamos este PPI por  $(A, b, c)$ . As restrições do tipo  $x \in \mathbb{Z}^n$  são denominadas de *restrições de integralidade*. Um PPI em que todas as suas soluções viáveis satisfazem a inequação  $x \leq 1$  é denominado de *problema de programação inteira 0-1*. O PPL obtido a partir do problema de programação inteira pela retirada de todas as restrições de integralidade é chamado de *relaxação linear* do PPI. Como o conjunto viável do PPI está contido no conjunto viável de sua relaxação linear, obtemos diretamente a seguinte proposição:

**Proposição 7.** *Se  $x$  é uma solução de um PPI e  $x^*$  é solução ótima de sua relaxação linear, então  $c^T x^* \leq c^T x$ .*

Chamaremos de *poliedro associado* a um PPI o fecho convexo das suas soluções viáveis. Seja  $P$  o poliedro associado a um PPI  $(A, b, c)$  e  $Q \supseteq P$ . Seja  $x_Q^*$  a solução ótima de  $(Q, c)$ . Se  $x_Q^* \in Q \setminus P$ , denominamos de *corte* de  $x_Q^*$  uma inequação  $a^T x \leq \alpha$ , válida para  $P$ , tal que  $a^T x_Q^* > \alpha$ . O *método de planos-de-corte* consiste em encontrar uma seqüência de conjuntos  $Q_0, \dots, Q_k$  tais que

$$Q_0 \supset Q_1 \supset \dots \supset Q_k \supseteq P$$

tais que  $Q_{i+1}$  é obtido a partir de  $Q_i$  pela inclusão de cortes sobre a solução ótima  $x_{Q_i}^*$ . Normalmente,  $Q_0$  é utilizado como sendo o conjunto viável da relaxação linear de  $(A, b, c)$ . É importante ressaltar que inequações que definem facetas de  $P$  são os cortes de melhor qualidade possível, pois estas inequações compõem uma descrição completa e irredundante do poliedro [19]. A idéia do *método de branch-and-cut* consiste em utilizar uma árvore de decisão do método branch-and-bound utilizando o método de planos-de-corte para gerar bons limitantes inferiores para os nós da árvore.

## Formulações de Programação Inteira para o Problema de Coloração

---

A importância do problema de coloração tem incentivado a investigação de métodos para encontrar limites inferiores próximos do número cromático, como apresentado em [30]. Historicamente, as primeiras tentativas para resolver este problema lidavam com o método de *branch-and-bound* com limites inferiores fornecidos por cliques maximais. Algumas formulações de programação linear surgiram ou foram estudadas como alternativa para obter limites inferiores melhores do que o valor de uma provável clique máxima [9, 20, 18, 37].

Neste capítulo apresentamos algumas formulações de programação linear para o problema de coloração e discutimos sobre as vantagens e desvantagens destas formulações. A Seção 3.1 traz uma discussão sobre parâmetros importantes a serem avaliados por uma boa formulação de programação linear. As seções a seguir trazem uma avaliação destes conceitos para algumas formulações. As seções 3.2, 3.3 e 3.4 têm os estudos associados às formulações de Conjuntos Independentes Maximais, Cores por Vértice e Orientações Acíclicas, respectivamente.

### 3.1. Parâmetros de Comparação

Alguns parâmetros de comparação podem ser usados para uma avaliação comparativa das formulações. Estes parâmetros de comparação serão a quantidade de variáveis e de restrições, o valor de  $C_5$ , a existência ou não de soluções viáveis que representam colorações simétricas e a qualidade das facetas existentes. A seguir, discutiremos um pouco sobre a importância destes parâmetros para um modelo de coloração.

**Quantidade de variáveis e restrições** A quantidade de variáveis e restrições definem o tamanho do modelo de programação linear. Dentre outros fatores, o tempo da solução de um modelo linear tem forte influência do seu tamanho. Para alguns modelos lineares, a quantidade de variáveis ou de restrições pode ser exponencial. Normalmente estes modelos precisam de um tempo adicional considerável para gerar variáveis ou restrições, quando necessário. Um parâmetro de avaliação destas formulações é o tamanho do modelo, por possuir uma descrição polinomial ou exponencial no tamanho do grafo.

**Valor de  $C_5$**  Pelo Teorema Forte dos Grafos Perfeitos [1, 13], todo grafo  $G$  possui uma clique de tamanho  $\chi(G)$ , um buraco ímpar ou anti-buraco ímpar como subgrafo induzido. Por este resultado, pode ser observado que subestruturas de  $G$  têm um papel importante no valor do número cromático.

Se um grafo  $G$  não possui buracos ímpares ou anti-buracos ímpares, este é perfeito e  $\chi(G)$  é igual ao tamanho de sua maior clique. Por isso, é importante que as formulações consideradas consigam colorir estas estruturas de forma eficiente. Chamaremos o *valor de  $C_5$*  de uma formulação como sendo o valor de sua relaxação linear aplicada ao grafo  $C_5$ . Caso  $G$  tenha buracos ou anti-buracos tendo uma forte influência em sua coloração, formulações com o maior valor de  $C_5$  provavelmente serão melhores em colori-lo. Chamaremos a *razão de aproximação de  $\chi_F(C_5)$*  o valor  $c^T x^*(C_5) / \chi_F(C_5)$ , onde  $c$  é o vetor de coeficientes de custo e  $x^*(C_5)$  é uma solução ótima da relaxação linear aplicada ao grafo  $C_5$ .

**Relação do valor ótimo da relaxação com  $\omega(G)$**  Sabemos que  $\omega(G)$  é um limite inferior para  $\chi(G)$ . É interessante que os limites inferiores gerados pelas formulações sejam maiores ou iguais ao tamanho de uma clique máxima. Caso contrário, heurísticas de geração de cliques grandes seriam um melhor limite inferior do que o valor da relaxação linear de tal modelo. Pode ser estudado esta relação com o modelo inicial ou se existem facetadas que geram este comportamento.

**Existência de soluções simétricas** Soluções baseadas em permutações de cores tornam-se desinteressantes por gerar grande simetria nas soluções. Para o problema de coloração, as formulações serão analisadas por dois tipos de simetria. Seja  $x'$  uma solução viável para uma formulação  $\Pi$  que representa uma coloração  $W$ . Caso a formulação  $\Pi$  possua uma solução  $x'' \neq x'$  que também represente  $W$ , as soluções viáveis  $x'$  e  $x''$  são ditas *soluções simétricas*. Neste caso, é dito que  $\Pi$  possui *simetria*. Caso  $x'$  e  $x''$  não possuam multicoloração, esta é dita uma *simetria permutacional*. Caso a simetria seja gerada por uma multicoloração de pelo menos um vértice, esta é dita uma *simetria de multicoloração*.

## 3.2. Conjuntos Independentes

### 3.2.1. Formulação

Uma coloração de um grafo pode ser vista como uma cobertura dos vértices por conjuntos independentes maximais. Seja  $G = (V, E)$  um grafo e  $\mathcal{W}$  o conjunto de todos os seus conjuntos independentes maximais. Seja a variável binária  $x_w$ , para cada  $w \in \mathcal{W}$ , que vale 1 quando o conjunto independente  $w$  for escolhido para fazer parte da cobertura e 0 quando o conjunto não for escolhido. O problema de coloração pode ser formulado como a seguir [37]:

$$\text{Minimizar } \sum_{w \in \mathcal{W}} x_w \quad (3.1)$$

$$\begin{aligned} \text{Sujeito a } \sum_{\{w: v \in w\}} x_w &\geq 1, & v \in V & \quad (3.2) \\ x_w &\in \{0, 1\}, & w \in \mathcal{W} & \end{aligned}$$

A única restrição deste modelo, restrição (3.2), indica que o vértice  $v$  deve ser coberto por pelo menos um conjunto independente maximal  $w$ . Observe que o número de conjuntos independentes maximais pode ser exponencial levando ao número de variáveis desta formulação poder ser exponencial. Deve ser observado que o valor da relaxação linear desta formulação é igual a  $\chi_F(G)$  [35].

### 3.2.2. Resolvendo o Problema Linear

Para resolver o problema gerado pelo número exponencial de variáveis, o método de geração de colunas pode ser usado para resolver a relaxação linear desta formulação. Este método mantém um subconjunto das variáveis, uma base primal viável e verifica a otimalidade desta base calculando o menor custo reduzido associado a todas as variáveis. Seja  $\pi$  o vetor de variáveis duais associadas às restrições (3.2). O problema de encontrar uma variável com custo reduzido de valor negativo equivale ao problema de encontrar um conjunto independente maximal ponderado, com ponderação  $\pi_v$  dada ao vértice  $v \in V$ , com valor maior do que 1 [37]. Observe que o problema de identificar a otimalidade de uma base é um problema NP-Difícil.

No texto que discute esta implementação em [37], o algoritmo de cálculo de uma clique máxima, proposto por Khoury e Pardalos [33], é modificado para a geração de conjuntos independentes máximos ponderados. O algoritmo de geração de um conjunto independente máximo é baseado na seguinte idéia: dado um grafo  $G = (V, E)$  e um vértice  $v \in V$ , um conjunto independente máximo em  $G$  está contido em  $G[V \setminus \{v\}]$  ou é igual a um conjunto independente máximo de  $G[\bar{N}(v)]$  junto a  $v$ . Esta idéia pode ser utilizada

para gerar um algoritmo recursivo:

$$CIMP(G) = \begin{cases} \max\{CIMP(G[V \setminus \{v\}]), \{v\} \cup CIMP(G[\bar{N}(v)])\}, & \text{se } V \neq \emptyset \\ \emptyset, & \text{se } V = \emptyset \end{cases}$$

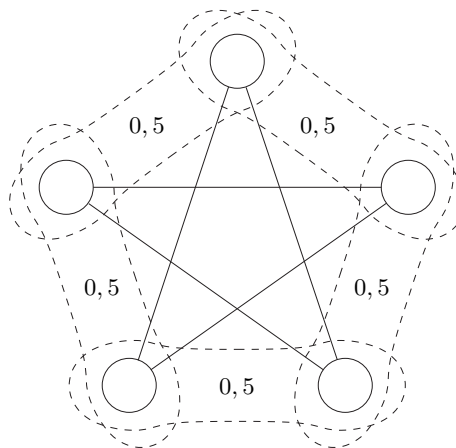
Para melhorar o desempenho desta enumeração, é observada a importância da ordem em que os vértices são utilizados para a recursão, baseada em um conjunto independente maximal guloso e na quantidade de vizinhos dos vértices. Outras melhorias utilizadas incluem a condição de parada do algoritmo quando algum conjunto independente violado é encontrado, identificação de subproblemas que não podem gerar soluções violadas e a utilização de outros métodos para identificar a solução ótima em subproblemas pequenos. Estas idéias aceleram ou evitam a solução completa do problema de obter um conjunto independente máximo ponderado a cada verificação por novas colunas mas, mesmo assim, a solução completa do problema de decisão deve ser obtido pelo menos uma vez, para provar que não existe nenhum conjunto independente violado.

### 3.2.3. Discussão

Esta formulação tem somente  $n$  restrições, porém um número exponencial de variáveis. O estudo desta formulação não inclui um estudo do poliedro associado. Isto é atribuído ao fato de que o método recomendado para tratar o número exponencial de variáveis é o de geração de colunas para resolver a sua relaxação, que não é incorporado com gerações de cortes tão facilmente. Além disto, esta formulação possui simetria de multicoloração ao usar conjuntos independentes maximais ao invés de conjuntos independentes não maximais.

Resultados indicam que  $\chi_F(G)$  é um limite inferior mais efetivo do que  $\omega(G)$  em uma implementação, mesmo considerado o tempo adicional necessário para o seu cálculo [37]. O número cromático fracionário regularmente supera o valor da clique máxima quando o tamanho da maior clique difere do número cromático. Por exemplo, o seu valor de  $C_5$  é 2,5 e a solução associada ao ótimo pode ser observada na Figura 3.1. O seguinte resultado mostra uma relação entre o limite inferior gerado e o valor da maior clique de  $G$ .





**Figura 3.1.** Solução da relaxação de CIM aplicada ao  $c_5$ . Os conjuntos independentes foram circutados com as linhas tracejadas e o valor das variáveis correspondentes são apresentados na figura.

**Lema 8.** Seja  $\pi(\text{CIM}, G)$  o valor da relaxação linear da formulação CIM aplicada ao grafo  $G$ . Para qualquer grafo  $G$ ,  $\pi(\text{CIM}, G) \geq \omega(G)$ .

*Prova:* Seja  $G = (V, E)$  um grafo e  $C$  uma clique máxima de  $G$ . Para cada  $v \in C$ , seja  $\mathcal{W}_v = \{w \in \mathcal{W} \mid v \in w\}$ . Note que, para  $u, v \in C$ , com  $u \neq v$ ,  $\mathcal{W}_u \cap \mathcal{W}_v = \emptyset$ . Assim, comparando a soma das restrições (3.2), para  $v \in C$ , com a função objetivo (3.1), obtemos o resultado do lema.  $\square$

Uma implementação deste método provou ser robusta para resolver instâncias com algumas centenas de vértices. Mesmo que a solução da sua relaxação linear envolva a solução de múltiplas instâncias de conjunto independente máximo ponderado, o tempo para resolver a árvore *branch-and-bound* foi muito bom comparado com uma implementação do método DSATUR [4], mostrando a força de seu limite inferior. A simetria da formulação foi tratada ao modificar a inequação (3.2) para uma igualdade e incluindo conjuntos independentes não maximais em  $\mathcal{W}$  [21]. Mesmo com a remoção de simetria e algumas outras modificações, os resultados computacionais não puderam ser melhorados.

Com o trabalho publicado em [37], Mehrotra e Trick mostraram que  $\chi_F(G)$  é um limite inferior muito mais efetivo que  $\omega(G)$ . Alguns desafios podem ser lançados a partir deste trabalho. Um deles corresponde a aproximar  $\chi_F(G)$  de forma mais eficiente. Outro possível desafio seria de obter um limite inferior para  $\chi(G)$  que seja superior a  $\chi_F(G)$ .

### 3.3. Cores por Vértice

#### 3.3.1. Formulação

Esta formulação é baseada na formulação inteira clássica para o problema de coloração com algumas restrições adicionais para remover parte da simetria. Seja  $w_i$ , para todo  $i \in \{1, \dots, n\}$ , uma variável binária tal que  $w_i = 1$  se, e somente se, a cor  $i$  é escolhida para ser usada na coloração. Seja  $x_{vi}$ , para todo  $v \in V$  e  $i \in \{1, \dots, n\}$ , tal que  $x_{vi} = 1$  se, e somente se, o vértice  $v$  recebe a cor  $i$ . Com estas variáveis, podemos construir a seguinte formulação [18]:

$$\begin{aligned} & \text{Minimizar } \sum_{j=1}^n w_j \\ & \text{Sujeito a } \sum_{j=1}^n x_{vj} = 1, \quad v \in V \end{aligned} \quad (3.3)$$

$$x_{uj} + x_{vj} \leq w_j, \quad uv \in E, j \in \{1, \dots, n\} \quad (3.4)$$

$$w_j \leq \sum_{v \in V} x_{vj}, \quad j \in \{1, \dots, n\} \quad (3.5)$$

$$w_j \geq w_{j+1}, \quad j \in \{1, \dots, n-1\} \quad (3.6)$$

$$x_{vj} \in \{0, 1\}, \quad v \in V, j \in \{1, \dots, n\}$$

$$w_j \in \{0, 1\}, \quad j \in \{1, \dots, n\}$$

A restrições (3.3) asseguram que cada vértice  $v$  deve receber exatamente uma cor. Restrições do tipo (3.4) asseguram que pares de vértices adjacentes recebem cores distintas. Esta restrição também garante que, caso a cor  $j$  seja utilizada por algum vértice  $v$ , então

$w_j = 1$ . As restrições (3.5) e (3.6) foram adicionadas ao modelo para remover simetrias. As restrições (3.5) garantem que uma cor  $j$  só será utilizada caso algum vértice  $v$  utilize a cor  $j$  e as restrições (3.6) asseguram que somente as cores iniciais serão utilizadas.

### 3.3.2. Facetas Baseadas em Subestruturas

Esta seção apresentará algumas facetas do poliedro associado à formulação de Cores por Vértices baseadas em subestruturas do grafo. As facetas apresentadas aqui estão relacionadas a buracos e a cliques do grafo. As seguintes inequações definem facetas e são baseadas em cliques [18]. Estas indicam que uma cor não pode ser atribuída a dois vértices diferentes de uma clique.

**Lema 9.** *Sejam  $j_0 \in \{1, \dots, n-1\}$  e  $C$  uma clique maximal. A inequação  $\sum_{u \in C} x_{uj_0} - w_{j_0} \leq 0$  define uma faceta.*

**Lema 10.** *Sejam  $\{v_1, \dots, v_p\}$  uma clique de tamanho  $p$ ,  $k \in \{p, \dots, n-1\}$  e  $Col \subseteq \{1, \dots, k-1\}$  com  $|Col| = p-1$ . A inequação*

$$\sum_{i=1}^p \sum_{j=k}^n x_{v_i j} + \sum_{i=1}^p \sum_{j \in Col} x_{v_i j} \leq w_k + \sum_{j \in Col} w_j$$

*é válida. Se  $\chi(G) + 1 \leq k$ , então esta inequação define uma faceta.*

A seguinte inequação é aplicada para buracos como subestruturas [18]:

**Lema 11.** *Sejam  $\{v_1, \dots, v_k\}$  um buraco, em  $G$ , de tamanho  $k$  e  $\{j_1, \dots, j_{k-1}\} \subseteq \{1, \dots, n-1\}$ . A inequação de buracos*

$$x_{v_1 j_1} + \sum_{i=2}^{k-1} (x_{v_i j_{i-1}} + x_{v_i j_i}) + x_{v_k j_{k-1}} + x_{v_k j_1} - w_{j_1} \leq k - 2$$

*é válida. Se uma  $\chi(G)$ -coloração ocorre em igualdade na inequação e existe no máximo um  $j_i$  tal que  $j_i \geq \chi(G) + 1$ , esta inequação define uma faceta.*

### 3.3.3. Discussão

Esta formulação é compacta ao possuir somente  $O(n^2)$  variáveis e  $O(nm)$  restrições. Além disto, esta contém facetos para cliques e buracos em geral, funcionando para buracos pares ou ímpares. O valor de  $C_5$  desta formulação é de 2. Uma observação que pode ser feita sobre a faceta de buracos é que não há como aplicá-la para que o valor da sua relaxação linear aumente aplicada ao grafo  $C_5$ .

Um ponto negativo desta formulação é que esta é bastante fracionária. Este termo pode ser observado no lema a seguir.

**Lema 12.** *Seja  $\pi(CPV, G)$  o valor da relaxação linear da formulação CPV aplicada ao grafo  $G$ . Para qualquer grafo  $G$ ,  $\pi(CPV, G) \leq 2$ .*

*Prova:* Para provar esta afirmação, basta observar que a seguinte coloração, com valor objetivo igual a 2 é válida para qualquer grafo  $G$ . Seja  $w_1 = 1$ ,  $w_2 = 1$  e  $w_j = 0$ , para  $3 \leq j \leq n$ . Para qualquer  $v \in V$ , faça com que  $x_{v1} = 0,5$ ,  $x_{v2} = 0,5$  e  $x_{vj} = 0$ , para  $3 \leq j \leq n$ . □

Este lema mostra, também, que esta formulação não captura os aspectos intrínsecos das estruturas de clique, buracos e anti-buracos. Sabemos que estas estruturas tem uma grande influência no número cromático e, por mais complexa que seja a estrutura de um grafo, o valor de sua relaxação linear será no máximo igual a 2.

Mesmo que as restrições (3.5) e (3.6) diminuam a simetria da formulação, por existir uma identificação de um valor para a cor, ainda existe simetria permutacional. Seja  $S_1, \dots, S_k$  uma  $k$ -coloração de  $G$  tal que o conjunto independente  $S_i$  é representado pela variável  $w_i$ , na formulação. Se  $\{i_1, \dots, i_k\}$  é uma permutação não trivial de  $\{1, \dots, k\}$ , então  $S_{i_1}, \dots, S_{i_k}$ , com  $S_{i_k}$  representado pela variável  $w_{i_k}$ , é uma outra maneira de descrever esta mesma coloração.

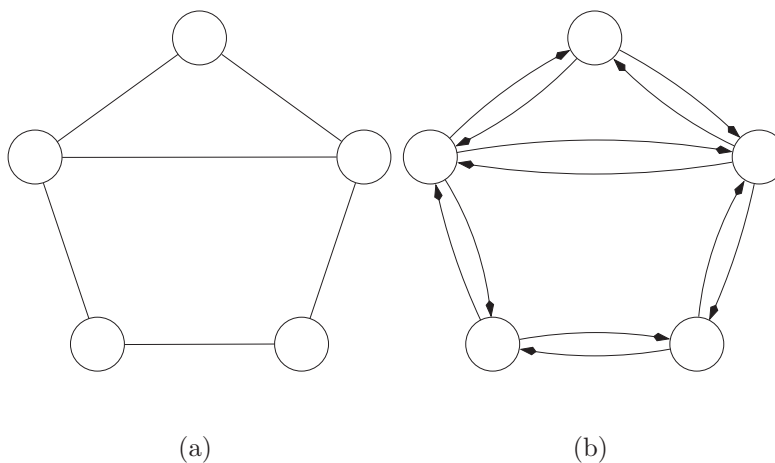
### 3.4. Orientações Acíclicas

#### 3.4.1. Formulação

Esta formulação é baseada no trabalho de Deming [17] em que o número cromático é comparado com as orientações acíclicas de um grafo  $G$  da seguinte maneira:

$$\chi(G) = \min_{G_\sigma \in \Omega(G)} \max_{\vec{p} \in \mathcal{P}_\sigma} |\vec{p}| + 1$$

onde  $\Omega(G)$  é o conjunto de todas as orientações acíclicas de  $G$  e  $\mathcal{P}_\sigma$  é o conjunto de todos os caminhos em  $G_\sigma$ .



**Figura 3.2.** Aplicação do operador  $D$ . O grafo direcionado apresentado em 3.2(b) é o resultado da aplicação do operador  $D$  sobre o grafo apresentado em 3.2(a)

Seja  $G = (V, E)$  um grafo não direcionado. Para utilizar este resultado, primeiro é necessário descrever o conjunto  $\Omega(G)$  de todas as orientações acíclicas de  $G$  em uma formulação de programação inteira 0-1. Defina o grafo direcionado  $D(G) = (V, A)$ , onde  $A = \{\vec{ij}, \vec{ji} \mid ij \in E\}$ . Um exemplo da aplicação deste operador pode ser encontrado na Figura 3.2. Para cada arco  $\vec{ij} \in A$ , defina uma variável binária  $w_{ij}$ , para indicar a direção desta aresta em uma orientação de  $G$ , com o seguinte significado:

se  $\sigma(ij) = j$  então  $w_{ij} = 1$  e  $w_{ji} = 0$

se  $\sigma(ij) = i$  então  $w_{ij} = 0$  e  $w_{ji} = 1$

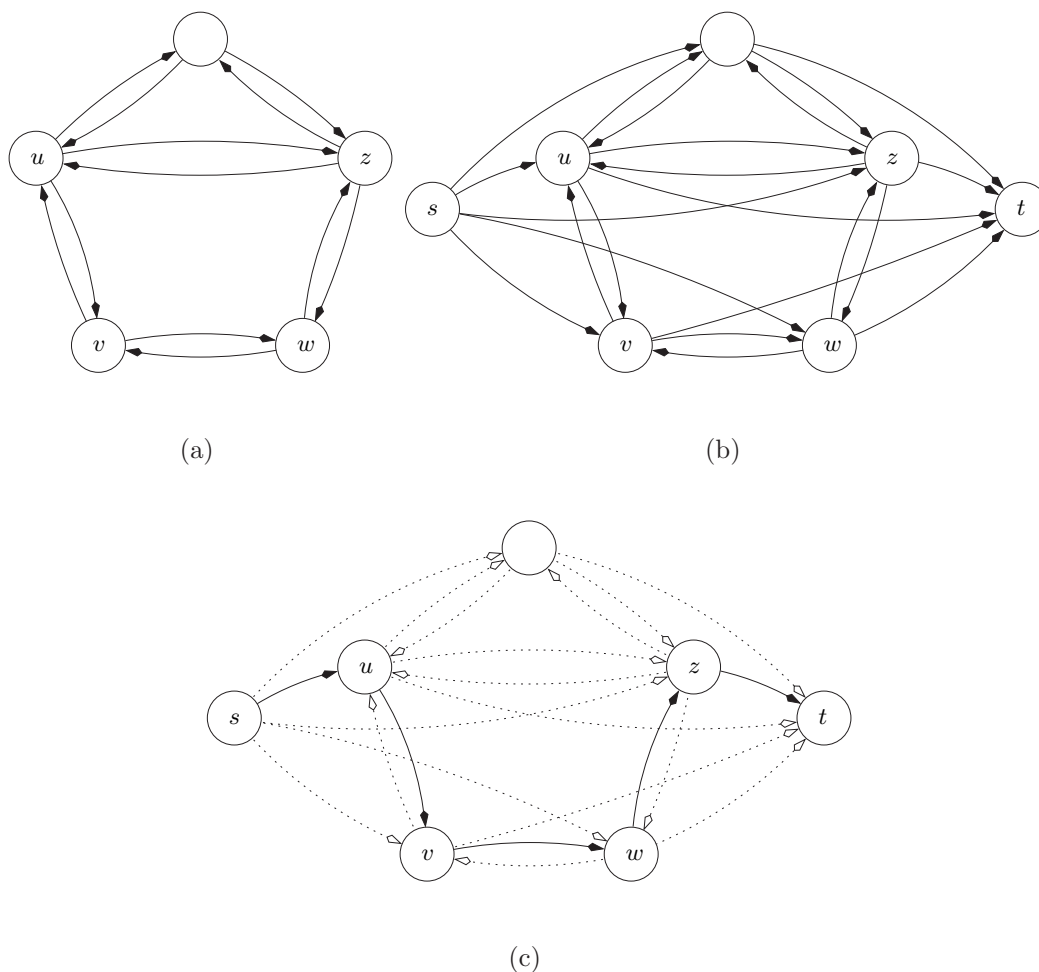
As seguintes restrições definem o conjunto de todas as orientações acíclicas de  $G$ :

$$\begin{aligned} w_{ij} + w_{ji} &= 1, ij \in E \\ \sum_{\vec{ij} \in \vec{C}} w_{ij} &\leq |\vec{C}| - 1, \text{ sendo } \vec{C} \text{ um ciclo direcionado em } D(G) \\ w_{ij} &\in \{0, 1\}, \vec{ij} \in A \end{aligned}$$

Para descrever um caminho máximo no grafo direcionado  $D(G)$ , defina os conjuntos  $V' = V \cup \{s, t\}$  e  $A' = A \cup \{\vec{sv}, \vec{vt} \mid v \in V\}$  e o grafo  $D'(G)$ , com os vértices  $s$  e  $t$  representando uma fonte e um sumidouro, respectivamente. Utilizamos um fluxo unitário de  $s$  a  $t$  para representar um caminho em  $D(G)$ . Tanto a construção do grafo  $D'(G)$  quanto a relação de um fluxo unitário com um caminho podem ser verificados no exemplo dado pela Figura 3.3. Este fluxo pode ser descrito com as seguintes restrições:

$$\begin{aligned} \sum_{\vec{ji} \in A'} x_{ji} - \sum_{\vec{ij} \in A'} x_{ij} &= 0, i \in V \\ \sum_{\vec{sj} \in A'} x_{sj} &= 1 \\ x_{ij} &\leq w_{ij}, \vec{ij} \in A \\ x_{ij} &\in \{0, 1\}, \vec{ij} \in A' \end{aligned}$$

Seja  $\vec{P}$  o conjunto de todos os caminhos no grafo direcionado  $D(G) = (V, A)$ . Para escolher um caminho de tamanho máximo, as seguintes restrições podem garantir esta propriedade:



**Figura 3.3.** Um grafo  $D(G)$  gerado pelo operador  $D$  pode ser observado na Figura 3.3(a). A Figura 3.3(b) mostra a construção do grafo  $D'(G)$ . A Figura 3.3(c) exibe um fluxo unitário de  $s$  a  $t$  que passa por  $u$ ,  $v$ ,  $w$  e  $z$ . O fluxo é representado pelos arcos escuros enquanto os arcos que não são percorridos pelo fluxo estão pontilhados. Este fluxo representa o caminho direcionado de  $u$  para  $z$  que passa por  $v$  e  $w$ .

$$\sum_{\vec{ij} \in A} x_{ij} \geq |\vec{p}| \left( \sum_{\vec{ij} \in \vec{p}} w_{ij} - |\vec{p}| + 1 \right), \vec{p} \in \vec{P} \quad (3.7)$$

Assim, o problema de coloração de grafos pode ser descrito como minimizar  $\sum_{\vec{ij} \in A} x_{ij}$  sujeito às restrições descritas acima [20].

### 3.4.2. Discussão

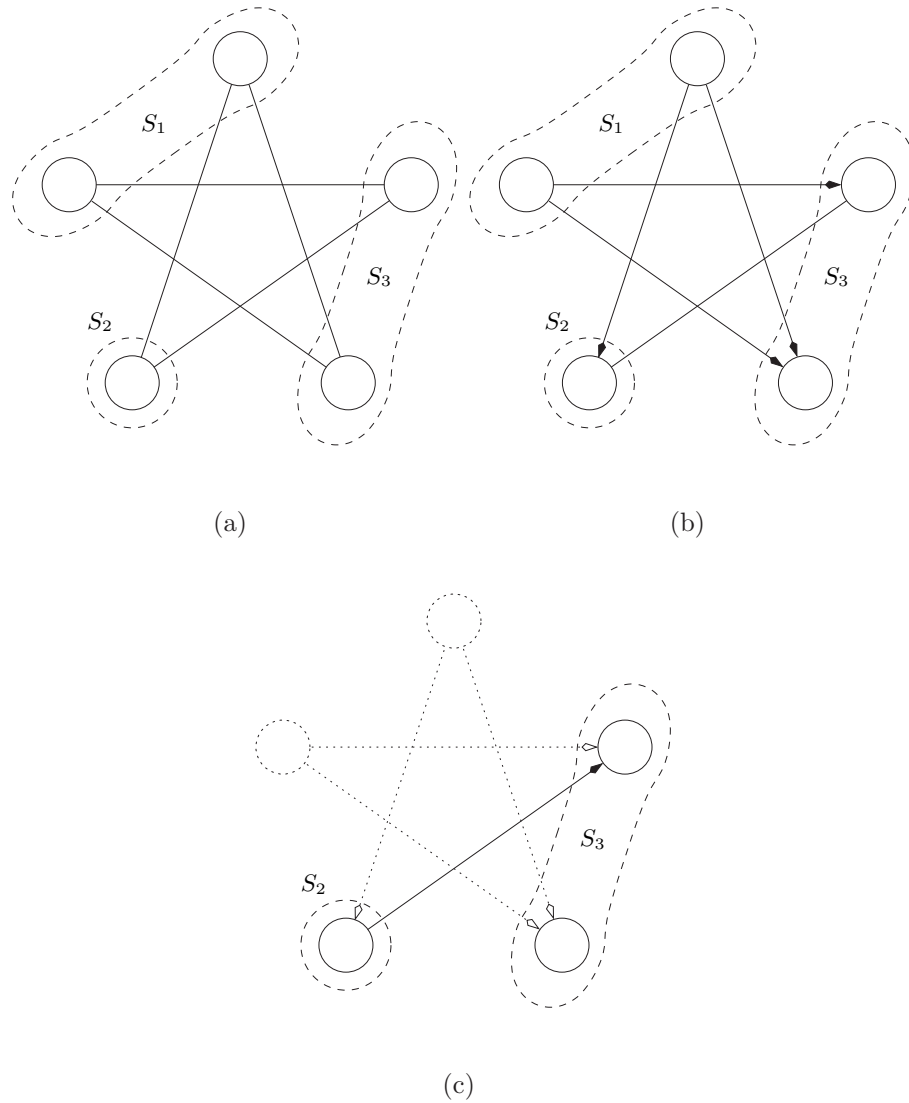
Esta formulação possui somente  $O(n + m)$  variáveis, porém, um número exponencial de restrições. A quantidade exponencial de restrições decorre das inequações (3.7) e de ciclo. Para uma descrição completa do modelo, seria necessário uma destas para cada caminho direcionado de  $D(G)$ . Esta formulação não contém facetas para cliques ou buracos. As únicas facetas expostas são relacionadas à estrutura da formulação. Sem as facetas, esta formulação não consegue capturar facilmente os aspectos intrínsecos das estruturas do grafo na coloração. Por exemplo, o valor de  $C_5$  desta formulação é 0,5, mesmo quando feita a descrição completa com todas as restrições do modelo.

Seja  $S_1, \dots, S_k$  uma  $k$ -coloração de  $G$ . É possível criar uma orientação acíclica que descreve esta coloração ao orientar para fora as arestas incidentes em  $S_j$  no grafo  $G[V \setminus \cup_{i=1}^{j-1} S_i]$ . Um exemplo deste método para gerar uma orientação acíclica do grafo pode ser encontrada na Figura 3.4. Para cada  $(i_1, \dots, i_k)$ , permutação não-trivial de  $(1, \dots, k)$ , podemos usar este método de geração de colorações em  $S_{i_1}, \dots, S_{i_k}$  para gerar outras soluções viáveis simétricas à primeira. Exemplos de duas soluções viáveis que definem a mesma coloração podem ser encontrados na Figura 3.5. Assim, esta formulação possui simetria permutacional.

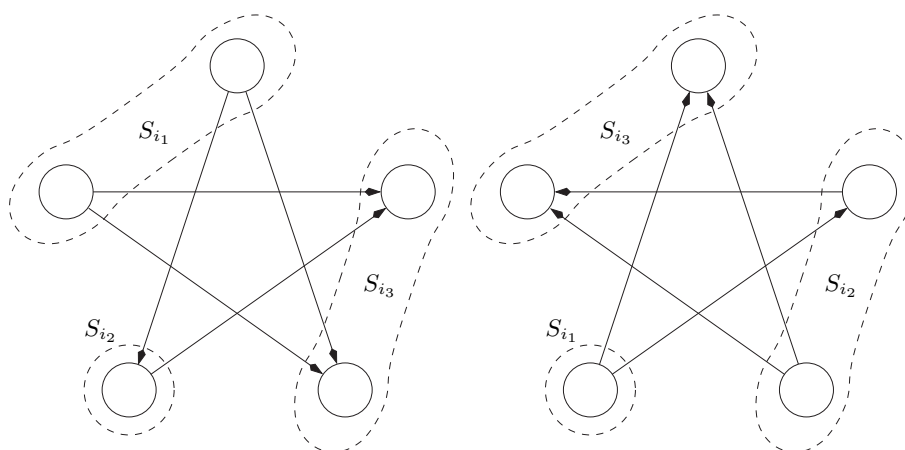
## 3.5. Tabela de Parâmetros das Formulações

Apresentamos uma tabela com os parâmetros discutidos entre as formulações na tabela 3.1. Para cada formulação, são apresentadas as seguintes informações:





**Figura 3.4.** Geração de uma orientação acíclica que define uma coloração  $S_1, S_2, S_3$ . Na Figura 3.4(a), apresentamos uma coloração do grafo  $C_5$ . Orientamos para fora de  $S_1$  as arestas que incidem sobre os seus vértices na Figura 3.4(b).



**Figura 3.5.** Duas soluções viáveis simétricas, geradas a partir da mesma coloração. Estas soluções viáveis foram geradas a partir da permutação  $(i_1, i_2, i_3)$ , como indicado na figura.

**Formulação** Formulação abordada. No caso da formulação REP, seus parâmetros serão apresentados no Capítulo 4.  $REP(\prec)$  representa a formulação dos representantes com orientação dos vértices.  $REP(\prec, =)$  representa a formulação dos representantes com orientação dos vértices e a remoção de multicolorações.

**Vars** Número de variáveis.

**Rest** Número de restrições.

**lb** Alguma informação sobre o limite inferior fornecido pela sua relaxação linear. Caso este campo esteja marcado com um asterisco, este resultado depende algum outro fator. Tais fatores são discutidos na seção de discussão sobre a respectiva formulação.

$C_5$  Valor de  $C_5$  desta formulação.

**Sim** Existência de soluções simétricas na formulação. As simetrias de multicoloração, permutação de cores e permutação de representantes são denotadas por M, P e R, respectivamente.

Formulação	Vars	Rest	lb	$C_5$	Sim
CIM [37]	Exp	$n$	$\chi_F(G)$	2,5	M
CPV [18]	$O(n^2)$	$O(mn)$	$\leq 2$	2,0	P
OA [20]	$O(n + m)$	Exp	-	0,5	P
REP [9]	$O(n + \bar{m})$	$O(mn)$	-	2,5	M+R
REP( $\prec$ ) [8]	$O(n + \bar{m})$	$O(mn)$	$\geq \omega(G)^*$	2,5	M
REP( $\prec, =$ )	$O(\bar{m})$	$O(mn)$	$\geq \omega(G)^*$	2,5	-

**Tabela 3.1.** Parâmetros comparativos sobre as formulações.

## Formulação dos Representantes

---

Neste capítulo, expomos a formulação que foi implementada neste trabalho. Esta formulação foi apresentada em [9] e consiste em nomear vértices para serem os representantes das classes de cor de uma coloração de  $G$ . Primeiro, mostramos a idéia por trás da formulação junto com uma formulação mais recente com menos simetria [8]. Como contribuição pessoal, destaco o estudo do poliedro desta formulação, cujos resultados podem ser encontrados em [8]. Alguns destes resultados sobre o estudo do politopo associado exibindo as facetas conhecidas são apresentados. Também discutimos sobre as vantagens e desvantagens desta formulação como fizemos no Capítulo 3 para as outras formulações.

Este capítulo está organizado da seguinte maneira. Na Seção 4.1 expomos a formulação por programação inteira 0-1. Em seguida, fazemos um estudo por facetas do politopo associado na Seção 4.2. A Seção 4.3 traz uma discussão sobre os fatores de comparação entre as formulações.

### 4.1. Formulação

Uma  $k$ -coloração de  $G = (V, E)$  pode ser descrita por um conjunto  $R = \{r_1, \dots, r_k\}$  de vértices, sendo cada vértice o *representante* da cor  $i$ , e uma família  $W_1, \dots, W_k$  de conjuntos independentes de  $G$ , sendo cada  $W_i \subseteq \bar{N}(r_i)$  na família formado pelos vértices

que recebem a mesma cor de  $r_i$ . Os vértices de  $W_i$  são ditos *representados* por  $r_i$ . Supondo uma ordem  $\prec$  definida sobre os vértices de  $G$ , podemos considerar que se  $u \prec v$ , então  $v$  não pode representar a cor de  $u$ . A ordem  $\prec$  induz uma orientação  $\sigma$  em  $\bar{G}$  tal que, para cada aresta  $uv$ ,  $\sigma(uv) = v$  se  $u \prec v$ . Baseados nessa orientação, usamos  $G^-(v)$  para representar  $G[\bar{N}^-(v)]$  e  $G^+(v)$  para representar  $G[\bar{N}^+(v)]$ . Sendo acíclica, a orientação  $\sigma$  produz dois conjuntos especiais não vazios: o conjunto de vértices fonte  $S = \{s \in V \mid \bar{N}^-(s) = \emptyset\}$  e o conjunto de vértices sumidouro  $T = \{t \in V \mid \bar{N}^+(t) = \emptyset\}$ . Note que um vértice  $u \in S$  não pode ser representado por qualquer outro vértice, o que significa que  $S \subseteq R$ . Podemos observar os conceitos de coloração por representantes segundo o exemplo apresentado na Figura 4.1.

Para descrever uma formulação de programação inteira, defina uma variável binária  $x_{uv}$ , para todo  $v \in V \setminus S$  e  $u \in \bar{N}^-[u]$ , com a seguinte interpretação:  $x_{uv} = 1$  se, e somente se,  $u$  representa uma cor de  $v$  caso  $v \neq u$  e  $u$  é representante de uma cor caso  $v = u$ . Para escrever a formulação em termos matemáticos, adotaremos a notação  $x(u, H) = \sum_{v \in H} x_{uv}$  e  $x(J, u) = \sum_{v \in J} x_{vu}$ , para quaisquer  $H \subseteq \bar{N}^+[u]$  e  $J \subseteq \bar{N}^-[u]$ . Caso  $H$  seja um único vértice  $v$  ou as extremidades de uma aresta  $vw$ , utilizaremos  $x(u, v)$  ou  $x(u, vw)$ , respectivamente. Para simplificar a descrição do modelo, utilizamos  $x(u, u) = 1$  quando o vértice  $u$  pertencer a  $S$ .

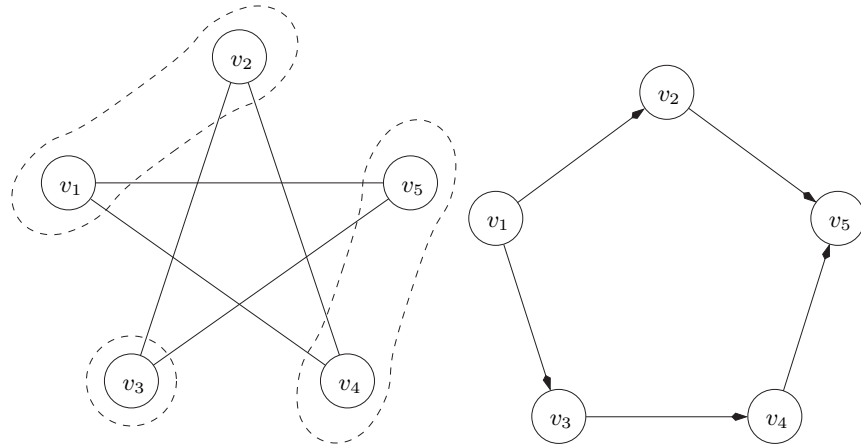
Um vetor  $x \in \mathbb{R}^{n+m-|S|}$  que contém as variáveis binárias associadas às possíveis representações entre pares de vértices é um vetor de incidência de uma coloração de  $G$  se, para todo  $u \in V \setminus S$ ,

$$x(\bar{N}^-[u], u) \geq 1 \tag{4.1}$$

e, para todo  $u \in V \setminus T$  e todo  $K \subseteq \bar{N}^+(u)$  que induz uma aresta ou uma clique maximal de tamanho 1 em  $G^+(u)$ ,

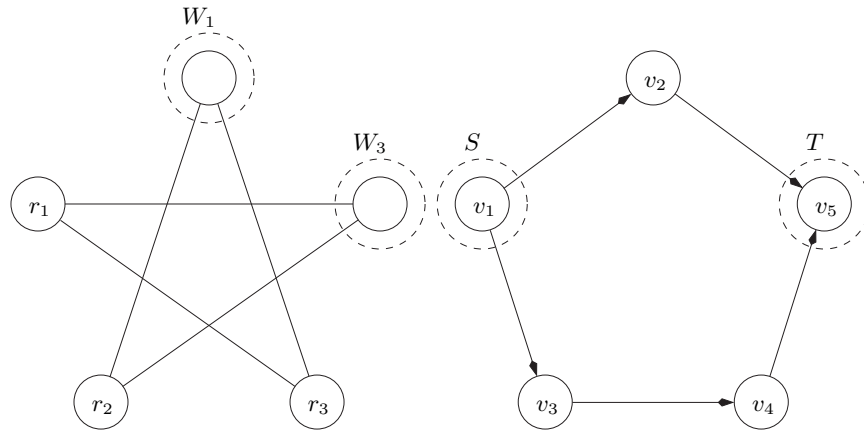
$$x(u, K) \leq x(u, u) \tag{4.2}$$

A inequação (4.1) indica que cada vértice  $u \in V \setminus S$  deve ser representado por ele mesmo ou por algum vértice na sua anti-vizinhança negativa. Como as extremidades de cada aresta devem receber cores diferentes, as inequações (4.2) asseguram que eles terão representantes



(a) Definimos a ordem  $\prec$  tal que, se  $i < j$ , então  $v_i \prec v_j$ . As classes de cor de uma coloração deste grafo estão circuladas.

(b) A orientação de  $\bar{G}$  segundo a ordem  $\prec$ .



(c) Coloração segundo a idéia dos representantes. Observe que o conjunto  $W_2$  é vazio.

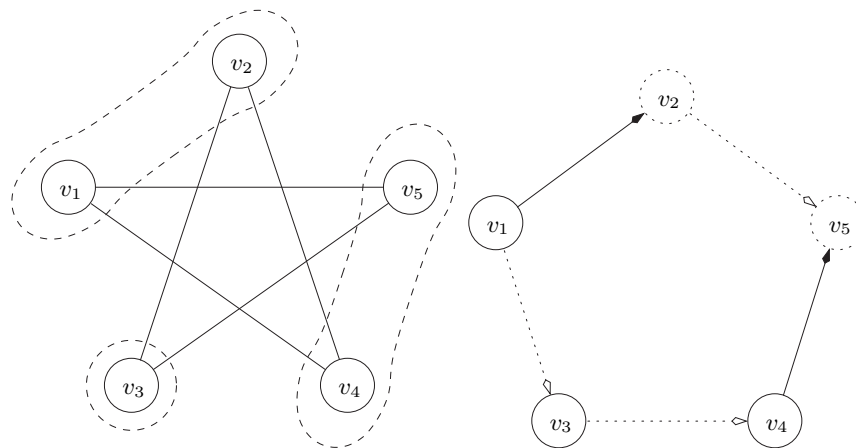
(d) Os conjuntos especiais  $S$  e  $T$  observados a partir de  $\bar{G}$ .

**Figura 4.1.** Definimos uma orientação e uma coloração para o grafo  $G = C_5$  na Figura 4.1(a). Na Figura 4.1(b), exibimos a orientação de  $\bar{G}$ . A coloração utilizando os conceitos de representantes é apresentada na Figura 4.1(c) e os conjuntos de vértices fonte e sumidouro na Figura 4.1(d).

distintos. Esta inequação também garante que um vértice  $v$  só pode ser representado por um vértice  $u$  se  $x(u, u) = 1$ . Um vetor de incidência que minimiza o número de cores

$$|S| + \sum_{u \in V \setminus S} x_{uu} \quad (4.3)$$

é uma coloração ótima de  $G$  [8]. A coloração apresentada na Figura 4.2 exibe o uso das variáveis apresentadas.



(a) Definimos a ordem  $\prec$  tal que, se  $i < j$ , então  $v_i \prec v_j$ . As classes de cor de uma coloração deste grafo estão circuladas.

(b) Utilizamos um vértice  $v$  com circunferência escura para denotar  $x(v, v) = 1$  e com circunferência pontilhada para denotar  $x(v, v) = 0$ . Utilizamos a mesma notação de escuro e pontilhado para um arco  $uv$ , denotando o valor de  $x(u, v)$ .

**Figura 4.2.** Descrevemos uma coloração na Figura 4.2(a). A representação desta coloração utilizando as variáveis do modelo dos representantes é apresentada na Figura 4.2(b).

## 4.2. Facetas e Politopo Associado

Nesta seção apresentamos resultados sobre o poliedro associado à formulação vista na seção anterior. As demonstrações destes resultados podem ser encontradas no Anexo A ou em [6].

O primeiro resultado sobre o poliedro consiste em definir a dimensão do espaço e quais restrições do modelo definem facetas. Para isto, considere  $P(G)$  como sendo o fecho convexo dos pontos  $x \in \{0, 1\}^{n+\bar{m}-|S|}$  que satisfazem as restrições (4.1) e (4.2).

**Teorema 13.**  $P(G)$  tem dimensão plena e as seguintes restrições definem facetas de  $P(G)$ :

$$x_{uu} \leq 1, \text{ para cada } u \in V \setminus S \quad (4.4)$$

$$x_{uv} \geq 0, \text{ para cada } u \in V \setminus T \text{ e } v \in \bar{N}^+(u) \text{ com } |\bar{N}^-(v)| \geq 2 \quad (4.5)$$

$$x_{uu} \geq 0, \text{ para cada } u \in T \quad (4.6)$$

$$x(\bar{N}^-[u], u) \geq 1, \text{ para cada } u \in V \setminus S \quad (4.7)$$

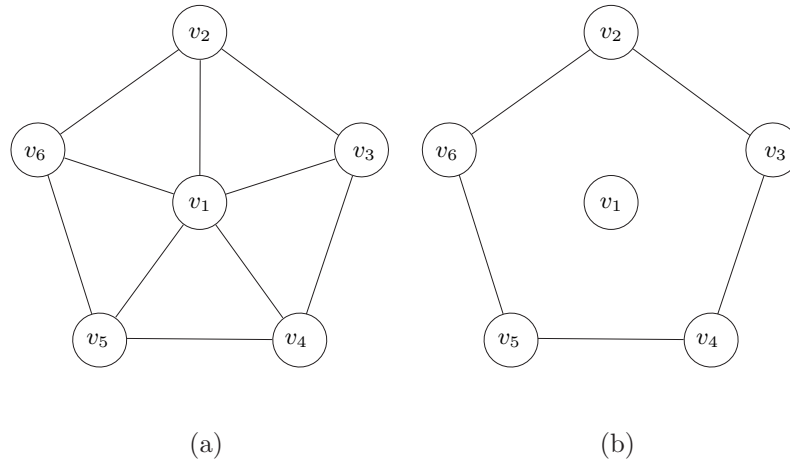
**Facetas Externas** As facetas descritas no teorema a seguir envolvem um vértice  $u \in V$  e a maior quantidade de vértices que podem ser representados por ele em um subconjunto  $H \subseteq \bar{N}^+(u)$ . Alguns termos especiais associados a  $H$  são utilizados. Para  $v \in H$ , denote por  $\alpha_v \in \mathbb{N}$  o tamanho máximo de um conjunto independente, de  $G[H]$ , que contém  $v$ . O máximo destes valores, para todo  $v \in H$ , é denotado por  $\alpha_H$ . Uma aresta  $vw$  é dita segura se existem conjuntos independentes  $W_v$  e  $W_w$ , ambos de  $G[H]$  e de mesmo tamanho, tais que

$$W_v \setminus W_w = \{v\}, W_w \setminus W_v = \{w\} \text{ e } \alpha_z = |W_v| = |W_w|, \forall z \in W_v \cup W_w. \quad (4.8)$$

Sejam  $E^{\text{safe}}$  o conjunto de arestas seguras de  $G[H]$  e  $G^{\text{safe}} = (H, E^{\text{safe}})$ . Na Figura 4.3(a), apresentamos o grafo  $W_5$ . Observe que os conjuntos independentes  $\{v_2, v_5\}$  e  $\{v_3, v_5\}$  fazem com que a aresta  $v_2v_3$  seja segura. O grafo  $W_5^{\text{safe}}$  pode ser observado na Figura 4.3(b). Para este grafo, temos  $\alpha_1 = 1$  e  $\alpha_2 = \dots = \alpha_6 = 2$ .

Uma generalização da inequação (4.2) é apresentada a seguir.





**Figura 4.3.** Ilustração dos conceitos de aresta segura e  $G^{\text{safe}}$ .

**Teorema 14.** Sejam  $u \in V \setminus T$  e  $H \subseteq \bar{N}^+(u)$ . A inequação externa

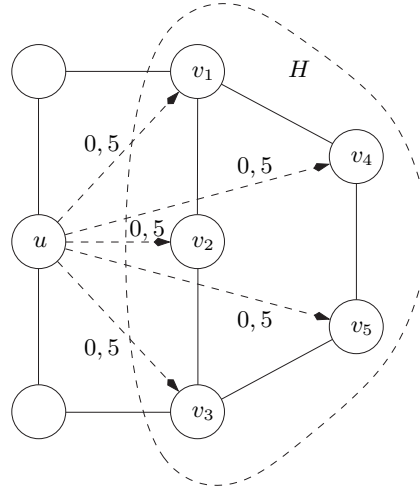
$$\sum_{v \in H} \frac{1}{\alpha_v} x(u, v) \leq x(u, u)$$

é válida. Esta define uma faceta se as seguintes propriedades forem válidas:

1.  $G[H]$  é  $\alpha_H$ -maximal em  $G^+(u)$ , o que significa que, se  $\bar{N}^+(u) \supseteq H' \supset H$ , então  $\alpha_{H'} > \alpha_H$  e
2. toda componente conexa  $C \subseteq G^{\text{safe}}$  contém um conjunto independente  $W$  tal que  $|W| = \alpha_v$ , para todo vértice  $v \in W$

É interessante ressaltar que a propriedade 2 é válida para muitos subgrafos que estão fortemente relacionados ao problema de coloração, sendo alguns deles clique, buracos ímpares, anti-buracos ímpares, rodas ímpares com centro e rodas pares sem centro com  $4k$  vértices. Podemos observar a inequação externa sendo violada para o caso de um buraco ímpar na Figura 4.4.

**Facetas Internas** Em contraste com a inequação anterior, agora analisamos inequações derivadas do número mínimo de representantes existentes em uma subestrutura induzida



**Figura 4.4.** Uma inequação interna violada com  $H = \{v_1, \dots, v_5\} \subseteq \bar{N}^+(u)$ .

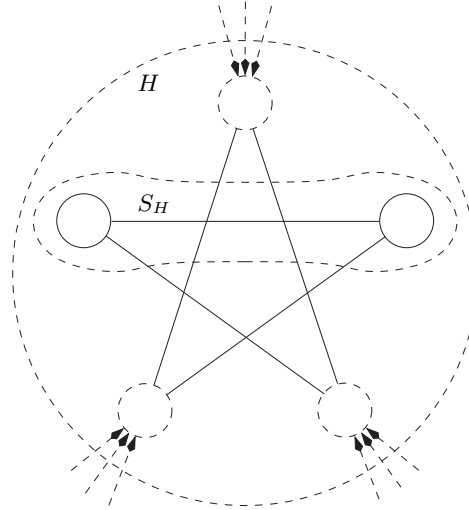
por um conjunto  $H \subseteq V$ . Para isto, defina  $S_H$  como sendo o conjunto de vértices fonte do grafo  $G[H]$ . Note que  $S_H$  é o conjunto  $\{s \in V(H) \mid \bar{N}_{G[H]}^-(s) = \emptyset\}$ , que não tem relação com o conjunto de vértices fonte de  $G$ . Observe que  $\chi(G[H])$  é o menor número de cores em  $H$  para colorir  $G[H]$  em  $G$ . Há exatamente  $|S_H|$  cores em  $G[S_H]$ . Portanto, deve haver pelo menos  $\chi(G[H]) - |S_H|$  cores em  $H \setminus S_H$ . Dessas cores,  $\sum_{v \in H \setminus S_H} x(v, v)$  são representadas por vértices em  $H \setminus S_H$  e  $\sum_{v \in H \setminus S_H} x(\bar{N}^-(v) \setminus H, v)$  são representadas por vértices fora de  $H$ . Assim, chegamos ao seguinte resultado.

**Teorema 15.** Se  $H \subseteq V$ , então a inequação interna

$$\sum_{v \in H \setminus S_H} x((\bar{N}^-(v) \setminus H) \cup \{v\}, v) \geq \chi(G[H]) - |S_H| \quad (4.9)$$

é válida para  $P(G)$ . A inequação interna define uma faceta se  $H$  induz um buraco ímpar ou um anti-buraco ímpar em  $G$ .

Para o grafo  $C_5$ , as variáveis envolvidas no somatório da inequação (4.9) podem ser esquematizadas como na Figura 4.5.



**Figura 4.5.** Esquema das variáveis envolvidas no somatório da inequação interna. As variáveis envolvidas são referentes aos vértices e arcos tracejados.

**Conjuntos Independentes** Seja  $H_0 \subseteq V \setminus S$  um conjunto independente de  $G$  com  $|H_0| \geq 2$ . Sejam  $v_0$  o vértice minimal de  $H_0$  com respeito a  $\prec$  e  $f(v_0)$  um vértice de  $H_0 \setminus \{v_0\}$ . Segue das inequações (4.1) que uma das seguintes condições é válida em qualquer coloração de  $G$ :

1.  $H_0$  contém pelo menos dois representantes, ou
2.  $v_0$  é representante e representa  $f(v_0)$ , ou
3.  $v_0$  ou  $f(v_0)$  são representados por anti-vizinhos que não estão em  $H_0$ .

Conseqüentemente, uma inequação válida é

$$\Phi(H_0) = \sum_{v \in H_0} x(v, v) + x(v_0, f(v_0)) + \sum_{v \in \{v_0, f(v_0)\}} x(\bar{N}^-(v) \setminus H_0, v) \geq 2. \quad (4.10)$$

Podemos melhorar a inequação (4.10) para definir uma faceta de  $P(G)$  da seguinte maneira. Seja  $\{v_1, \dots, v_\ell\}$ , para  $\ell > 0$ , um conjunto de vértices cuja união com  $H_0$  constitui um conjunto independente de  $G$ . Suponha que  $v_i \prec v_{i-1}$ , para todo  $i \in \{1, \dots, \ell\}$ .

Para  $H_i = H_0 \cup \{v_1, \dots, v_i\}$  e  $f(v_i) \in H_0$  defina, recursivamente, a seguinte inequação:

$$\Phi(H_i) = \Phi(H_{i-1}) + x(v_i, \{v_i, f(v_i)\}) - x(v_i, \{v_0\} \cup f(H_{i-1})) + \Delta_i \geq 2, \quad (4.11)$$

onde  $f(H_{i-1}) = \cup_{j=0}^{i-1} \{f(v_j)\}$  e

$$\Delta_i = \begin{cases} x(\bar{N}^-(f(v_i)) \setminus H_i, f(v_i)), & \text{se } f(v_i) \notin f(H_{i-1}) \cup \{v_0\}, \\ 0, & \text{caso contrário.} \end{cases}$$

Podemos escrever esta inequação de forma não recursiva da seguinte maneira:

$$\Phi(H_i) = \sum_{v \in H_i} x(v, v) + \sum_{j=0}^i x(v_j, f(v_j)) + \sum_{v \in f(H_i) \cup \{v_0\}} x(\bar{N}^-(v) \setminus H_i, v). \quad (4.12)$$

Ainda podemos fortalecer a inequação (4.12) criando a inequação

$$\Phi'(H_i) = \Phi(H_i) - \sum_{v \in H_i \setminus (f(H_i) \cup \{v_0\})} x(v, K_v) \geq 2$$

onde  $K_v$  é uma clique maximal no grafo gerado pelos vértices do conjunto

$$Q_v = \{z \in \bar{N}^+(v) \mid N(z) \supseteq f(H_i) \cap \bar{N}^+(u) \setminus F(u)\}.$$

onde  $F(u) = \{f(u)\}$  se  $u \notin H_0$  e  $F(u) = \emptyset$  caso contrário. A partir de  $\Phi'(H_i)$ , podemos obter um resultado forte sobre esta inequação, denominada de inequação de conjuntos independentes.

**Teorema 16.** *A inequação de conjuntos independentes  $\Phi'(H_i) \geq 2$  é válida para  $P(G)$ , para todo  $i \in \{1, \dots, \ell\}$ , dado que  $H_i \cap S = \emptyset$ . Ela define uma faceta se, e somente se, as duas condições a seguir são satisfeitas:*

1.  $|f(H_i) \setminus \{v_0\}| \geq 2$ ,
2. os dois maiores vértices de  $H_0$  com respeito a  $\prec$  pertencem a  $f(H_i)$

### 4.3. Discussão

Mesmo com a ordenação dos vértices para remover parte da simetria existente, esta formulação ainda contém simetria de multicoloração. Esta simetria ocorre pela multicoloração permitida na restrição (4.1). Uma solução seria transformar esta inequação em uma igualdade. Com isto, poderíamos remover as variáveis  $x_{uu}$  com a transformação

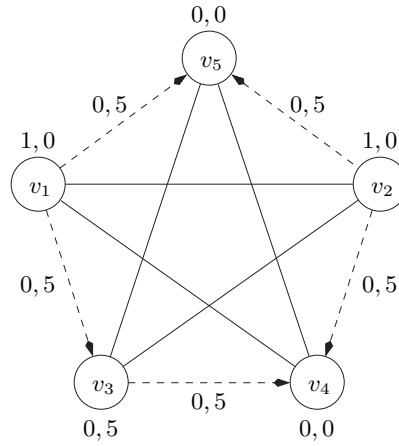
$$x_{uu} = 1 - x(\bar{N}^-(u), u). \quad (4.13)$$

Devido aos limites da variável  $x_{uu}$ , podemos substituir a restrição (4.1) por

$$x(\bar{N}^-(u), u) \leq 1. \quad (4.14)$$

Assim, obtemos um modelo sem simetrias e mais compacto que o anterior, contendo somente  $\bar{m}$  variáveis e  $O(nm)$  restrições. A partir de agora, ainda utilizaremos o valor  $x(u, u)$ . Porém, isto não implica que a variável  $x_{uu}$  esteja no modelo. O uso será para denotar o valor da expressão  $1 - x(\bar{N}^-(u), u)$ .

O valor de  $C_5$  desta formulação é de 2,5 e uma solução que define este valor pode ser vista na Figura 4.6. Porém, este valor é aumentado para 3 com a inclusão da inequação interna aplicada ao grafo  $C_5$ . As facetas desta formulação se dividem nas inequações internas, externas e de conjuntos independentes. As inequações externas fazem com que um vértice  $u$  não possa representar mais vértices do que um conjunto independente maximal em  $H \subseteq \bar{N}^+(u)$ . Esta define faceta quando  $H$  é dado por muitos subgrafos que estão fortemente relacionados ao problema de coloração, como por exemplo cliques, buracos ímpares, anti-buracos ímpares, rodas ímpares com centro e rodas pares sem centro com  $4k$  vértices. Pode ser observado a partir de [8] que se utilizarmos somente as inequações externas, o limite inferior fornecido pela relaxação linear do modelo também é um limite inferior para o número cromático fracionário. As inequações internas impõem um número mínimo de cores para  $G[H]$ . Esta inequação define uma faceta para buracos ímpares e anti-buracos ímpares, estruturas onde é difícil obter um bom limite inferior com formulações de programação inteira.



**Figura 4.6.** Uma solução ótima que define o valor de  $C_5$  desta formulação. As arestas do grafo estão representadas por linhas escuras enquanto as variáveis referentes aos à orientação do complemento estão representadas pelas setas tracejadas. O valor indicado em um vértice  $u$  indica o valor de  $x(u, u)$  e o valor indicado em um arco  $uv$  indica o valor de  $x(u, v)$ .

Esta formulação não garante obter o valor de  $\chi(G)$  quando  $G$  é perfeito. Porém, há alguns casos em que podemos garantir que o limite inferior gerado será igual a  $\chi(G)$ . Para um dos casos, basta observar pela função objetivo (4.3), que  $|S|$  é um limite inferior para o problema. Caso seja conhecido  $K$ , uma clique máxima de  $G$ , a ordenação do grafo pode ser feita com  $u \prec v$  para quaisquer que sejam  $u \in K$  e  $v \in V \setminus K$ . Assim garantimos que  $S = K$  e a formulação fornece um limite inferior igual a  $\chi(G)$ . Para o resultado a seguir, para  $K \subseteq V$ , defina  $\bar{N}^-(K) = \cup_{u \in K} \bar{N}^-(u) \setminus K$ . O limite inferior ótimo também poderá ser obtido com uma quantidade suficiente de facetas adicionadas no modelo, como observado do seguinte resultado.

**Teorema 17.** Seja  $\pi(REP, G)$  o valor da relaxação linear da formulação REP aplicada ao grafo  $G$ . O valor de  $\pi(REP, G)$  será maior ou igual a  $\omega(G)$  se, para alguma clique  $K$ , com  $|K| = \omega(G)$ , a inequação  $x(\bar{N}^-(u), u) + x(u, \bar{N}^+(u) \cap K) \leq 1$  for válida, para  $u \in \bar{N}^-(K)$ .

*Prova:* Fazendo a transformação (4.13) sobre a função objetivo (4.3), obtemos a função objetivo

$$\begin{aligned}\pi(\text{REP}, G) &= n - \sum_{u \in V \setminus S} x(\bar{N}^-(u), u) \\ &= n - \sum_{u \in (V \setminus S) \cap K} x(\bar{N}^-(u), u) - \sum_{u \in \bar{N}^-(K) \setminus S} x(\bar{N}^-(u), u) \\ &\quad - \sum_{u \in V \setminus (S \cup K \cup \bar{N}^-(K))} x(\bar{N}^-(u), u)\end{aligned}$$

sem as variáveis  $x(u, u)$ . Podemos observar que, o primeiro destes somatórios equivale a  $\sum_{u \in \bar{N}^-(K)} x(u, \bar{N}^+(u) \cap K)$ . Majorando o valor de  $x(\bar{N}^-(u), u)$  por 1 no terceiro somatório, obtemos

$$\begin{aligned}\pi(\text{REP}, G) &\geq n - \sum_{u \in \bar{N}^-(K)} (x(u, \bar{N}^+(u) \cap K) + x(\bar{N}^-(u), u)) \\ &\quad - |V \setminus (S \cup K \cup \bar{N}^-(K))| \\ &\geq n - |\bar{N}^-(K)| - n + |S \cup K \cup \bar{N}^-(K)| \\ &= |(S \cup K) \setminus \bar{N}^-(K)|.\end{aligned}$$

Mas como  $K \cap \bar{N}^-(K) = \emptyset$ , obtemos o resultado desejado.  $\square$

Note que a faceta externa descrita no Teorema 14, quando aplicada a  $H$  sendo uma clique maximal e aplicando a transformação (4.13), é da forma  $x(\bar{N}^-(u), u) + x(u, H) \leq 1$ . Seja  $K$  uma clique máxima de  $G$ . Se para  $u \in \bar{N}^-(K)$ , a faceta externa for incluída com  $\bar{N}^+(u) \cap K \subseteq H$ , a inequação necessária para o Teorema 17 ser válido se aplicará a  $u$ . Assim, não é necessário conhecer uma clique máxima  $K$  para satisfazer as hipóteses deste teorema, pois estas inequações podem ser incluídas no modelo naturalmente como cortes do método de plano-de-corte.

Observe que podemos utilizar esta formulação junto com os resultados obtidos em [8] para resolver os desafios lançados por Mehrotra e Trick [37] que estão descritos na Seção 3.2.3. Para obter uma aproximação de  $\chi_F(G)$ , podemos utilizar esta formulação com as inequações externas. Para obter um limite inferior para  $\chi(G)$  que seja melhor do que  $\chi_F(G)$ , podemos utilizar esta formulação com as demais inequações apresentadas na Seção 4.2.

## Representantes por Planos-de-Corte Faciais

---

Durante a elaboração desta dissertação, foi implementado o cálculo de limitantes inferiores para o número cromático fracionário utilizando a formulação dos representantes. Esta implementação é baseada no método de planos-de-corte para reforçar a integralidade e melhorar os limitantes inferiores [15]. Somente facetas externas são consideradas. Por [7], sabemos que obtemos um limite inferior para  $\chi_F(G)$ .

Na implementação de um algoritmo por planos de corte, muitos fatores devem ser balanceados para obter bons resultados. Idéias para balancear uma implementação por planos de corte podem ser encontradas na literatura em alguns trabalhos [19]. Este capítulo inclui o estudo de muitos testes para minimizar o tempo de execução desta implementação.

Como estamos tentando resolver um problema computacionalmente difícil, diminuir o tamanho do problema de entrada antes da solução do problema ajuda a obter melhores tempos. A Seção 5.1 descreve os algoritmos de pré-processamento que serão aplicados antes da solução do problema. A seguir, na Seção 5.2, descreveremos formas de manter o tamanho do problema linear com uma quantidade aceitável de restrições. Na Seção 5.3, mostraremos heurísticas que foram implementadas para resolver o problema de separação. Enfim, a Seção 5.4 trará uma visão geral do algoritmo de planos de corte junto com a condição de parada.



## 5.1. Pré-processamento

Na solução de problemas computacionalmente difíceis, diminuir o tamanho da entrada, incluir informações para facilitar a solução ou quebrá-lo em problemas menores constitui boa prática para diminuir o tempo total de processamento. Nesta seção exibimos os algoritmos de pré-processamento que são aplicados a um grafo  $G$  para diminuir o seu tamanho ou particioná-lo em grafos menores. Na Seção 5.1.1, mostramos como remover um vértice  $u \in V$  tal que seja fácil encontrar uma coloração ótima para  $G$ , dado que é conhecida uma coloração ótima para  $G[V \setminus \{u\}]$ . Na Seção 5.1.2, descrevemos como particionar  $V$  em  $\{V_1, \dots, V_j\}$  de forma que a coloração ótima de  $G$  possa ser obtida facilmente pela coloração ótima de  $G[V_1], \dots, G[V_j]$ .

O algoritmo de pré-processamento consiste em duas etapas. A primeira aplica uma remoção de vértices descrita na Seção 5.1.1 para decompor  $G$  em um grafo  $G'$ . A segunda etapa consiste na aplicação do algoritmo de decomposição descrito na Seção 5.1.2 a  $G'$ .

### 5.1.1. Remoção de Vértices

Remover vértices apropriados do grafo original pode reduzir substancialmente o tempo necessário para resolver o problema, e isto pode ser obtido se observarmos que um vértice  $u \in V$  pode ser removido de  $G$  se pudermos obter uma coloração ótima de  $G$  a partir de uma coloração ótima de  $G[V \setminus \{u\}]$ . Nesta seção, identificamos algumas situações onde esta propriedade é válida e, conseqüentemente, diminuí o tamanho do grafo de entrada. Três situações em que vértices podem ser removidos são as seguintes:

1.  $u$  é um vértice universal: isto ocorre quando  $d(u) = n - 1$ , o que significa que a cor atribuída a  $u$  deve ser distinta das cores de todos os vértices em  $V \setminus \{u\}$ , em qualquer coloração de  $G$ . Conseqüentemente, a propriedade segue ao atribuir a  $u$  uma cor que não aparece em  $G[V \setminus \{u\}]$ .
2.  $u$  é dominado por outro vértice  $v$ : esta situação é caracterizada por  $N(u) \subseteq N(v)$ . Note que um vértice  $u$  é dominado por  $v$  se, e somente se,  $u$  tem grau 0 no subgrafo

induzido pela anti-vizinhança de  $v$ . Para provar esta afirmação, observe que, como cada vizinho de  $u$  também é vizinho de  $v$  (o que implica que  $u$  é anti-vizinho de  $v$ ),  $u$  é um vértice isolado em  $G[\bar{N}(v)]$ . Se  $u$  é isolado na anti-vizinhança de  $v$ , então  $u \in \bar{N}(v)$  e todos os vizinhos de  $u$  também são vizinhos de  $v$ . Usando esta caracterização, é fácil ver que, dada qualquer coloração ótima  $C$  de  $G[V \setminus \{u\}]$ , se  $u$  recebe a mesma cor que  $v$  em  $C$ , obtemos uma coloração para  $G$  com tantas cores quanto  $C$ .

3.  $u$  tem grau baixo: esta situação ocorre quando  $d(u) < \chi(G)$ . Neste caso, no máximo  $\chi(G) - 1$  cores aparecem na vizinhança de  $u$  em qualquer coloração ótima  $C$  de  $G[V \setminus \{u\}]$ . Assim, uma coloração ótima de  $G$  é obtida de  $C$  ao atribuir a  $u$  uma cor que não aparece em  $N(u)$ .

Algumas observações podem ser feitas sobre o algoritmo de remoção de vértices, apresentado no Algoritmo 1. Antes da sua execução, uma heurística é chamada para calcular, em tempo  $O(n^3)$ , um limite inferior  $\underline{\chi}$  para  $\chi(G)$  como o tamanho de uma provável clique máxima. Tal limite inferior é utilizado para identificar casos de vértices de grau baixo, como descrito na situação 3. Remover vértices dominados pode gerar novos vértices universais, dominados ou de grau baixo. Este é o motivo do algoritmo iterar até que vértices não sejam mais removidos. Em contrapartida, remover vértices universais não pode gerar novos vértices universais, dominados ou de grau baixo. Entretanto, o limite inferior para o grafo diminui em uma unidade, o que também ocorre quando um vértice de grau baixo com grau exatamente igual a  $\underline{\chi} - 1$  é removido. Estas idéias nos motivaram a percorrer os vértices em ordem não decrescente de grau, para verificar uma remoção. Outros percursos sobre os vértices foram testados, incluindo ordens aleatórias e em ordem não crescente de grau, e este se mostrou como a alternativa mais rápida. A cada percurso sobre a lista  $L$ , são encontrados vértices que podem ser removidos do grafo. Quando um vértice é removido do grafo, uma variável booleana *removed* é colocada em **true** para indicar que a lista deve ser percorrida novamente. Estes vértices e as suas respectivas vizinhanças no momento em que ele foi removido de  $G$  são armazenados em uma pilha  $R$ .

**Algoritmo 1.** *Remoção de vértices universais, dominados e de baixo grau*

```

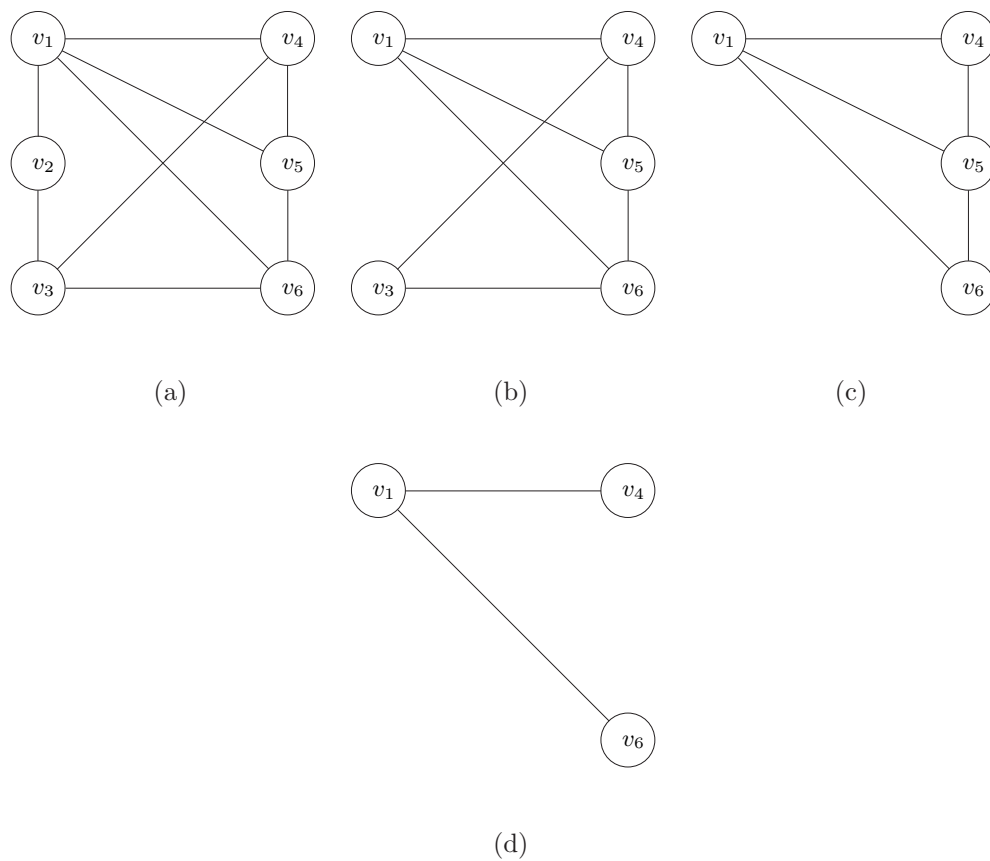
1:  $R \leftarrow \emptyset$ ;  $removed \leftarrow \mathbf{true}$ 
2: enquanto  $removed$  faça
3:    $removed \leftarrow \mathbf{false}$ 
4:    $L \leftarrow V$  em ordem não decrescente de grau em  $G$ 
5:   enquanto  $L \neq \emptyset$  faça
6:      $v \leftarrow$  remover elemento mínimo de  $L$ 
7:     se  $d(v) < \underline{\chi} - 1$  ou  $d(v) = |V| - 1$  então
8:        $removed \leftarrow \mathbf{true}$ ;  $R \leftarrow R \cup \{(v, N(v))\}$ ; Remover  $v$  de  $G$ 
9:     senão
10:      para todo  $u \in V$  faça
11:        se  $N(v) \subseteq N(u)$  então
12:           $removed \leftarrow \mathbf{true}$ ;  $R \leftarrow R \cup \{(v, N(v))\}$ ; Remover  $v$  de  $G$ 
13:        senão se  $d(v) = \underline{\chi} - 1$  então
14:           $removed \leftarrow \mathbf{true}$ ;  $R \leftarrow R \cup \{(v, N(v))\}$ ; Remover  $v$  de  $G$ 
15:           $\underline{\chi} \leftarrow \underline{\chi} - 1$ ;

```

A complexidade de tempo do Algoritmo 1 é dominada pela iteração sobre a lista  $L$ . No pior caso, o algoritmo percorre a lista  $n$  vezes, com complexidade de  $O(n \log n)$  para gerar esta lista a cada iteração e complexidade de  $O(n^3)$  para decidir a remoção de um vértice. Isto dá uma complexidade de pior caso de  $O(n^5 \log n)$ . Apesar desta complexidade de pior caso parecer ruim, este algoritmo costuma ter um número pequeno de percursos sobre a lista de prioridade, mesmo que muitos vértices do grafo sejam removidos e o algoritmo de detecção de vértices dominados também não costuma demorar um tempo proporcional à complexidade de pior caso. Também, note que um vértice dominado  $u \in V$  também pode ser um vértice de grau baixo e que a verificação de  $u$  ser dominado pode ser evitado caso este seja de grau baixo ou universal. Implementando a verificação dos vértices de forma que a última verificação seja do vértice ser dominado, este algoritmo de remoção de vértices costuma ser consideravelmente rápido mesmo para grandes instâncias de entrada e quando um grande número de vértices for removido.

A Figura 5.1 exhibe o comportamento do Algoritmo 1 em um exemplo. O grafo fornecido como entrada é exibido na Figura 5.1(a). Supomos que a lista ordenada gerada para percorrer os vértices seja  $L = \{v_2, v_3, v_5, v_1, v_4, v_6\}$  e que o limite inferior fornecido seja  $\underline{\chi} = 3$ . Inicialmente verificamos que o vértice  $v_2$  é dominado pelo vértice  $v_6$ . Ao remover  $v_2$  de  $G$ , obtemos o grafo apresentado na Figura 5.1(b). Em seguida, identificamos o vértice  $v_3$  como dominado por  $v_1$  e o removemos do grafo, obtendo o grafo apresentado na Figura 5.1(c). Observe que o vértice  $v_5$  se tornou um vértice universal. Com a remoção deste vértice, diminuimos o valor de  $\underline{\chi}$  para 2 e obtemos o grafo apresentado na Figura 5.1(d).

Para recolorir os vértices que foram removidos pelo Algoritmo 1, basta percorrer o conjunto  $R$  na ordem inversa das remoções que foram feitas. Para colorir um vértice  $v \in R$ , basta olhar as cores dos vértices que estão em sua vizinhança. É atribuída uma cor a  $v$  que foi utilizada para colorir  $G$  e que não se encontra em sua vizinhança. Caso tal cor não exista, uma nova cor é atribuída a  $v$ .



**Figura 5.1.** Exemplo da decomposição de um grafo.

### 5.1.2. Decomposição por Componentes Conexas e Componentes Conexas do Complemento

Nesta seção, particionamos  $V$ , o conjunto de vértices de  $G$ , em  $V_1, \dots, V_j$  de forma que a coloração ótima de  $G$  possa ser obtida facilmente pela coloração ótima de  $G[V_1], \dots, G[V_j]$ . Duas situações em que esta partição pode ser feita são as seguintes:

1.  $G[V_1], \dots, G[V_j]$  definem as componentes conexas de  $G$ . Sejam  $C_1, \dots, C_j$  colorações ótimas para estes subgrafos, com  $\chi_i^*$  cores sendo utilizadas na coloração  $C_i$ . Uma coloração ótima para  $G$  seria tal que  $\chi(G) = \max_{i \in \{1, \dots, j\}} \chi_i^*$  e pode ser construída fazendo com que as cores utilizadas nas componentes conexas sejam as mesmas utilizadas na componente que utiliza mais cores.
2.  $G[V_1], \dots, G[V_j]$  definem as componentes conexas de  $\bar{G}$ . Para quaisquer  $G[V_{i'}]$  e  $G[V_{i''}]$ , com  $i' \neq i''$ , um vértice  $u \in V_{i'}$  é adjacente a todos os vértices de  $V_{i''}$ . Isto implica que a cor atribuída a  $u$  não é utilizada para colorir qualquer vértice de  $V_{i''}$ . Sejam  $C_1, \dots, C_j$  colorações ótimas para estes subgrafos, com  $\chi_i^*$  cores sendo utilizadas na coloração  $C_i$ . Uma coloração ótima para  $G$  seria tal que  $\chi(G) = \sum_{i \in \{1, \dots, j\}} \chi_i^*$  e pode ser construída fazendo com que as cores utilizadas nestes subgrafos sejam distintas.

Caso  $\bar{G}$  seja conexo, aplicar a decomposição da situação 1 pode fazer com que  $\bar{G}[V_i]$  fique desconexo, para algum  $i$ . Caso  $G$  seja conexo, aplicar a decomposição da situação 2 pode fazer com que  $G[V_i]$  fique desconexo, para algum  $i$ . Assim, esta decomposição pode ser aplicada a  $G$  alternando testes entre a situação 1 e a situação 2 até que, para um determinado  $V_i \subseteq V$ , as situações 1 e 2 não possam ser aplicadas a  $G[V_i]$ . Esta idéia é apresentada no Algoritmo 2. A lista  $L$  representa uma lista de subgrafos tais que é necessário verificar uma das situações descritas. Esta verificação é feita a partir de  $G$  ou  $\bar{G}$  e um subconjunto de vértices, que indicará qual situação deve ser verificada e qual o subgrafo considerado. Considere a partição  $V_1^*, \dots, V_{j'}^*$  de  $V$  de forma que  $G[V_1^*], \dots, G[V_{j'}^*]$  não possa mais ser particionado segundo as situações descritas. Note que, caso  $G$  não

**Algoritmo 2.** *Decomposição de  $G$  por componentes conexas e componentes conexas do complemento*

- 1:  $L \leftarrow \emptyset; F \leftarrow \emptyset$
- 2: Decompor  $G$  usando o Algoritmo 1 obtendo  $G'$
- 3: **para todo**  $G'_j$  componente conexa de  $G'$  **faça**
- 4:    $L \leftarrow L \cup (V(G'_j), \bar{G}')$
- 5: **enquanto**  $L \neq \emptyset$  **faça**
- 6:    $(H, G'') \leftarrow$  remover elemento de  $L$
- 7:   **se**  $G''[H]$  for conexo **então**
- 8:      $F \leftarrow F \cup H$
- 9:   **senão**
- 10:     **para todo**  $G''_j$  componente conexa de  $G''[H]$  **faça**
- 11:        $L \leftarrow L \cup (V(G''_j), \bar{G}'')$

possua vértices dominados,  $G[V_i^*]$  também não terá vértices dominados, para qualquer  $i \in \{1, \dots, j'\}$ . Assim, começamos a nossa decomposição a partir da decomposição obtida pelo Algoritmo 1. A complexidade de tempo deste algoritmo é dominada pela decomposição obtida pelo Algoritmo 1, dado que o restante da decomposição pode ser feito em  $O(n^3)$ .

## 5.2. Tratando o Problema Linear

Discutimos algumas técnicas que usamos para manter o tamanho do problema linear em limites aceitáveis. Tais técnicas são necessárias pois o número de facetas no politopo associado à formulação dos representantes cresce exponencialmente com o número de vértices. O problema linear resolvido a cada iteração do método de planos-de-corte é chamado de *PL núcleo* ou simplesmente *núcleo*, quando for claro pelo contexto. Testes realizados indicam que, caso a quantidade de restrições não seja tratada, a estrutura do

núcleo torna-se complexa o suficiente para que o tempo necessário para resolvê-lo fique inaceitável e domine o tempo de processamento. Resultados melhores podem ser obtidos com um subconjunto das restrições no modelo e passos adicionais de solução deste modelo linear para decidir quais restrições devem se manter no modelo principal. Vale ressaltar que a formulação que tratamos é a formulação sem simetria de multicoloração descrito na Seção 4.3, com a remoção das variáveis  $x_{uu}$  pela transformação (4.13).

O modelo inicial associado à formulação dos representantes será descrito na Seção 5.2.1. Na Seção 5.2.2, descreveremos o critério de seleção de restrições para o modelo principal, um repositório de inequações e a política para manter restrições no repositório.

### 5.2.1. Modelo Inicial

Apresentamos o LP núcleo inicial utilizado para descrever uma coloração pela formulação dos representantes. Para obtê-lo, é necessário obter uma ordem  $\prec$  sobre o conjunto de vértices  $V$ . Esta ordem é obtida da seguinte maneira. Partimos de  $K'$ , uma provável clique máxima de  $G$ . A ordem  $\prec$  é uma qualquer de distâncias não decrescentes em  $G$  de algum vértice de  $K'$ . Note que esta ordem é tal que  $S = K'$ . Algumas observações podem ser feitas sobre a escolha de tal ordenação. Esta ordenação garante que o limite inferior fornecido é superior ou igual a  $|K'|$ . Investigações sobre ordens aleatórias, mais especificamente com  $|S| < |K'|$ , encontram limites inferiores menores do que  $|K'|$  com uma certa regularidade, o que reforça a escolha desta orientação inicial.

Testes realizados indicam que incluir uma descrição completa do modelo dos representantes faz com que o tempo de solução do método de planos de corte seja muito custoso. Assim, utilizamos um LP núcleo inicial mais compacto, mantendo apenas um subconjunto das inequações (4.2). Esta escolha decorre do fato de que muitas destas restrições não são utilizadas para definir uma solução ótima. Estas restrições iniciais pertencem ao PL núcleo durante todas as iterações de solução do método de planos-de-corte. As demais restrições que possam entrar no modelo podem entrar e sair segundo a discussão apresentada na Seção 5.2.2.



Os limites inferior e superior das variáveis  $x_{uv}$ , com  $u \in V \setminus T$  e  $v \in \bar{N}^+(u)$ , podem ser tratados junto ao núcleo sem acarretar restrições adicionais ao modelo. Assim, para cada uma delas, teremos os limites  $0 \leq x_{uv} \leq 1$ . Para cada  $u \in V \setminus T$ , seja  $\mathcal{K}_u = \{K_1, \dots, K_{j_u}\}$  uma cobertura de  $\bar{N}^+(u)$  em cliques maximais de  $G^+(u)$ . O núcleo contém, para cada  $u \in V \setminus (S \cup T)$  e  $i \in \{1, \dots, j_u\}$ , as restrições

$$x(\bar{N}^-(u), u) + x(u, K_i) \leq 1. \quad (5.1)$$

Temos a restrição (5.1) para  $u \in S$  e  $i \in \{1, \dots, j_u\}$  se, e somente se,  $|K_i| > 1$ . Para  $u \in T$ , se  $|\bar{N}^-(v)| > 1$ , colocamos a restrição

$$x(\bar{N}^-(u), u) \leq 1. \quad (5.2)$$

Observe que é possível que algumas restrições de aresta (4.2) sejam violadas em uma solução ótima. Estas restrições entram no núcleo somente quando ficarem violadas a partir da heurística de separação descrita na Seção 5.3.1. A escolha destas restrições garante duas propriedades importantes para os algoritmos de separação, a saber:

1. nenhum vértice será representado com um valor maior do que uma unidade; e
2.  $x_{uv} \leq x_{uu}$ , para todo  $u \in V \setminus T$  e  $v \in \bar{N}^+(u)$ .

### 5.2.2. Seleção de Inequações

Uma restrição é dita *ativa* se ela está no núcleo. Caso contrário, ela é dita *inativa*. Um repositório de restrições é utilizado para guardar algumas restrições inativas para que estas possam retornar ao núcleo. Utilizamos  $x^* \in \mathbb{R}^{\bar{m}}$  para denotar a *solução ótima corrente* do núcleo. Dados um PL núcleo, um repositório de restrições inativas e a solução  $x^*$ , dois problemas associados à seleção de inequações podem ser levantados. O primeiro corresponde a selecionar um subconjunto de inequações do núcleo que devem se tornar inativas. O segundo consiste em selecionar inequações inativas do repositório para se tornarem ativas.

Considerando o primeiro problema, defina o critério de decisão de quais restrições continuam no núcleo após encontrar uma solução ótima parcial  $x^*$  por *política de manutenção de restrições*. Uma política de manutenção de restrições que tem sido bastante utilizada na literatura é manter restrições em igualdade [19]. Mesmo cortando uma boa quantidade de restrições, esta solução se tornou impraticável por usar uma quantidade muito grande de restrições para descrever a solução ótima obtida na última solução do núcleo. Outra possibilidade para a política de manutenção de restrições é a de manter somente as restrições cuja variável dual associada é básica. Esta é uma forma mais compacta de descrever a solução ótima corrente para o núcleo pois os pontos de solução ótima para o núcleo sem a remoção das restrições e com a remoção baseada nesta política de manutenção são os mesmos e esta gera um núcleo com, no máximo, a mesma quantidade de restrições que seriam obtidas no núcleo caso a política de manutenção de restrições em igualdade fosse utilizada. Nesta política de manutenção, as restrições que definem o modelo inicial não serão verificadas para sair do núcleo para que as propriedades 1 e 2 da Seção 5.2.1 se mantenham sempre válidas, facilitando o trabalho das heurísticas de separação. Note que, para um grafo de entrada  $G$  e um modelo inicial  $\pi$ , a aplicação desta política de manutenção fará com que a descrição da solução corrente do núcleo tenha um número limitado de restrições.

Quando uma restrição se torna inativa, esta é adicionada ao repositório de restrições para que possa retornar, posteriormente, ao núcleo. Para evitar que o tamanho do repositório se torne excessivamente grande, a cada restrição que se encontra neste repositório é atribuída uma *idade inicial* e tem uma condição em que a restrição é considerada velha para ser descartada, também chamada de *condição de velhice*. Uma restrição envelhece quando o repositório é percorrido e esta não corta a solução atual do LP núcleo. Uma restrição volta à sua idade inicial quando esta passa do repositório para o núcleo.

A menos que seja especificado o contrário, as restrições terão um comportamento padrão de idade inicial, envelhecimento e condição de velhice. O padrão consiste na idade inicial sendo um inteiro com valor inicial 0. Ao envelhecer, o valor de sua idade é acrescido

em uma unidade e a condição de velhice ocorre quando a restrição tem um valor de idade igual a 10. Em outras palavras, o padrão é que restrições fiquem no máximo por 10 iterações no repositório sem que estas sejam descartadas. Este comportamento indica a idéia de que restrições que passam muito tempo sem cortar uma solução ótima corrente provavelmente não cortarão soluções futuras do núcleo.

Para resolver o segundo problema, dado que foram removidas restrições e conseguimos uma descrição compacta da solução ótima corrente pela política de manutenção de restrições, qualquer restrição que corte a solução ótima corrente será adicionada ao núcleo, sendo ela obtida do repositório ou por heurísticas de separação. Para muitos grafos, a quantidade de restrições que entra no núcleo tende a ser muito grande. A idéia por trás destas escolhas é que a quantidade de restrições que definem a solução atual seja mínima e todas as restrições adicionais cortem a solução ótima corrente. Esta escolha visa obter o maior passo possível entre o valor de duas soluções ótimas consecutivas do núcleo.

### 5.3. Heurísticas de Separação para Facetas Externas

O problema de separação consiste em obter uma inequação válida que corte a solução ótima corrente  $x^*$  do PL núcleo. Na Seção 5.3.1, tratamos o problema de separação relacionado a inequações externas aplicadas a cliques enquanto na Seção 5.3.2 tratamos o problema de separação relacionado a inequações externas aplicadas a buracos ímpares.

#### 5.3.1. Cliques Externas

Sejam  $u \in V \setminus T$  e  $K \subseteq \bar{N}^+(u)$  um subconjunto de vértices que induz uma clique maximal em  $G^+(u)$ . A inequação de clique  $Q_u(K)$ , associada a  $u$  e  $K$  é descrita por:

$$x(u, K) \leq x(u, u). \quad (5.3)$$

A base de  $Q_u(K)$  é o subconjunto

$$\mathcal{K}(Q_u(K)) = \{v \in K \mid 0 < x^*(u, v) < x^*(u, u)\}. \quad (5.4)$$

Em outras palavras, a base de  $Q_u(K)$  é formada pelos vértices de  $K$  que são parcialmente representados por  $u$ . Observe que  $Q_u(K)$  pode ter uma base vazia. Em tal caso,  $Q_u(K)$  não é violada por  $x^*$  ou  $K$  contém arestas violadas. A verificação deste fato é exibida na seguinte propriedade:

**Propriedade 18.** *A inequação de clique  $Q_u(K)$ , para a clique maximal  $K \subseteq \bar{N}^+(u)$ , é violada por  $x^*$  se, e somente se,*

1.  $x^*(u, \mathcal{K}(Q_u(K))) > x^*(u, u)$ ; ou
2. existe uma aresta  $vw \in E[K]$ , com  $v \notin \mathcal{K}(Q_u(K))$ , tal que  $x^*(u, vw) > x^*(u, u)$ .

*Prova:* Seja  $Q_u(K)$  violada. Suponha, por contradição, que as condições 1 e 2 do enunciado desta propriedade não sejam violadas. Segue que  $x^*(u, v) \leq x^*(u, vw) \leq x^*(u, u)$ , para todo  $v \in K$  e  $w \in K \setminus \{v\}$ . Se, para algum vértice  $v \in K$ , a equação  $x^*(u, v) = x^*(u, u)$  for válida, então  $x^*(u, w) = 0$ , para todo  $w \in K \setminus \{v\}$ . Isto implica que  $x^*(u, K) = x^*(u, v) = x^*(u, u)$ , que contradiz o fato de  $Q_u(K)$  ser violada. Assim,  $x^*(u, v) < x^*(u, u)$ , para todo  $v \in K$ . Consequentemente, podemos escrever  $K = \mathcal{K}(Q_u(K)) \cup \{v \in K \mid x^*(u, v) = 0\}$  e  $x^*(u, K) = x^*(u, \mathcal{K}(Q_u(K))) \leq x^*(u, u)$ , que também contradiz o fato de  $Q_u(K)$  ser violada.

Caso a condição 1 ou a condição 2 sejam válidas, claramente  $Q_u(K)$  será violada.  $\square$

Definimos a *anti-vizinhança fracionária de  $u$*  como sendo o conjunto  $\bar{N}_{1/2}^+(u) = \{v \in \bar{N}^+(u) \mid 0 < x^*(u, v) < x^*(u, u)\}$ . Como sugerido pela Propriedade 18, nossa heurística de separação procura por inequações clique violadas na anti-vizinhança reduzida de  $u$ , para cada  $u \in V \setminus T$ . Durante a construção de  $\bar{N}_{1/2}^+(u)$ , verificamos se existe uma aresta violada  $vw$ , para cada  $v \in \bar{N}^+(u) \setminus \bar{N}_{1/2}^+(u)$  e  $w \in \bar{N}^+(u) \cap N(v)$ . Esta busca é feita pois a descrição da formulação dada pelo nosso modelo inicial não é completa, como discutido na Seção 5.2.1.

A heurística de separação de cliques, apresentada no Algoritmo 3, é uma enumeração parcial de cliques maximais em  $\bar{N}_{1/2}^+(u)$ . Inicialmente, procuramos por arestas violadas,

**Algoritmo 3.** *Separação de cliques externas*

```

1: para todo  $u \in V$  tal que  $x^*(u, u) > 0$  faça
2:   Desmarcar todos os vértices de  $\bar{N}^+(u)$ ;  $\bar{N}_{1/2}^+(u) \leftarrow \emptyset$ 
3:   para todo  $v \in \bar{N}^+(u)$  faça
4:     se  $x^*(u, v) = x(u, u)$  então
5:        $f \leftarrow \text{false}$ 
6:       para todo  $w$  adjacente a  $v$  em  $G^+(u)$  que esteja desmarcado faça
7:         se  $x_{uv}^* + x_{uw}^* > x_u^*$  então
8:           Encontrar clique maximal  $K$  de  $G^+(u)$  que contém  $\{v, w\}$ 
9:           Adicionar  $x(u, K) \leq x_u$  ao LP núcleo
10:           $f \leftarrow \text{true}$ 
11:        se  $f$  então
12:          Marcar  $v$ 
13:        senão se  $x^*(u, v) > 0$  então
14:           $\bar{N}_{1/2}^+(u) \leftarrow \bar{N}_{1/2}^+(u) \cup \{v\}$ 
15:        se  $x^*(u, u) < x^*(u, \bar{N}_{1/2}^+(u))$  então
16:          para todo  $v \in \bar{N}_{1/2}^+(u)$  faça
17:             $\omega_v \leftarrow x(u, v)$ 
18:          para todo  $v \in \bar{N}_{1/2}^+(u)$  faça
19:            se  $v$  está desmarcado então
20:              Encontrar clique maximal  $K$ , ponderada por  $\omega$ , de  $G[\bar{N}_{1/2}^+(u) \cap N[v]]$ 
21:              se  $x^*(u, K) > x^*(u, u)$  então
22:                Adicionar  $x(u, K) \leq x_u$  ao LP núcleo
23:              Marcar todos os vértices em  $K$ 

```

quando  $u$  representa em mais de uma unidade dois vértices adjacentes  $v$  e  $w$ . Este caso representa a condição 2 da Propriedade 18. Em seguida, tentamos garantir a inclusão de pelo menos uma inequação clique para cada  $v \in \bar{N}_{1/2}^+(u)$ . Este segundo caso representa a condição 1 da Propriedade 18.

### 5.3.2. Buracos Externos

Nesta seção, para  $u \in V$ , utilizamos  $H$  e  $P_k$  como subconjuntos de  $\bar{N}^+(u)$ , com  $h$  e  $k$  vértices, que induzem, em  $G^+(u)$ , um buraco ímpar e um caminho, respectivamente. A inequação de buracos  $L_u(H)$ , associada a  $u \in V$  e  $H$  é dada por

$$x(u, H) \leq \frac{h-1}{2}x(u, u). \quad (5.5)$$

**Propriedade 19.** *Se a inequação de buracos  $L_u(H)$  é violada por  $x^*$ , então*

1.  $H \subseteq \bar{N}_{1/2}^+(u)$  ou
2. existe uma aresta  $vw \in E[H]$  tal que  $x^*(u, vw) > x^*(u, u)$ .

*Prova:* Seja  $H = \{v_1, \dots, v_h\}$  um subconjunto de  $\bar{N}^+(u)$  que induz um buraco ímpar em  $G$  tal que  $x^*(u, H) > \frac{h-1}{2}x^*(u, u)$  e que as arestas em  $E[H]$  não são violadas por representação de  $u$ . Por contradição, suponha que  $x^*(u, v) = 0$  ou  $x^*(u, v) = x^*(u, u)$ , para algum  $v \in H$ . Sem perda de generalidade, suponha que  $v = v_1$ . Se  $x^*(u, v_1) = 0$ , então  $x^*(u, H) = x^*(u, P_{h-1})$ , com  $P_{h-1} = \langle v_2, \dots, v_h \rangle$ . O somatório das inequações  $x^*(u, v_{2i}v_{2i+1}) \leq x^*(u, u)$  para  $i \in \{1, \dots, (h-1)/2\}$  resulta em  $x^*(u, H) = x^*(u, P_{h-1}) \leq \frac{h-1}{2}x^*(u, u)$ , que contradiz a hipótese. Caso  $x^*(u, v_1) = x^*(u, u)$ , temos que  $x^*(u, v_2) = 0$  por não haver arestas violadas por  $u$  e obtemos a mesma contradição de antes com  $v = v_2$ . Assim,  $0 < x^*(u, v) < x^*(u, u)$ , para todo  $v \in H$ .  $\square$

A heurística de separação proposta para buracos ímpares é baseada na seguinte propriedade.

**Propriedade 20.** Se  $L_u(H)$  é violada por  $x^*$ , então, para cada  $k \in \{1, \dots, h-1\}$ , existe  $P_k \subseteq H$  tal que  $x^*(u, P_k) > \frac{k}{2} \frac{h-1}{h} x^*(u, u)$ .

*Prova:* Seja  $H = \{v_0, \dots, v_{h-1}\}$ . Suponha que exista  $k \in \{1, \dots, h-1\}$  tal que, para todo  $P_k \subseteq H$ , a inequação  $x^*(u, P_k) \leq \frac{k}{2} \frac{h-1}{h} x^*(u, u)$  seja válida. Considere

$$P_k^i = \{v_i, \dots, v_{(i+k-1) \bmod h}\} \subseteq H, \text{ para } i \in \{0, \dots, h-1\}. \quad (5.6)$$

Obtemos que  $x^*(u, H) = \frac{1}{k} \sum_{i=0}^{h-1} x^*(u, P_k^i) \leq \frac{h}{k} \frac{k}{2} \frac{h-1}{h} x^*(u, u) = \frac{h-1}{2} x^*(u, u)$ , o que contradiz a hipótese de  $L_u(H)$  ser violado por  $x^*$ .  $\square$

**Corolário 21.** Se  $h \geq 5$ , então  $x^*(u, P_k) > 0, 4x^*(u, u)$ .

**Algoritmo 4.** Separação de buracos ímpares externos

- 1: **para todo**  $u \in V$  tal que  $x^*(u, u) > 0$  **faça**
- 2:     **para todo**  $G[B]$ , componente conexa de  $\bar{N}_{1/2}^+(u)$ , com  $|B| \geq 5$  **faça**
- 3:         **para todo**  $v \in B$  tal que  $x^*(u, v) > 0, 4x^*(u, u)$  **faça**
- 4:             **para todo**  $wz \in \bar{E}[B \cap N(v)]$  tal que  $x^*(u, \langle w, v, z \rangle) > 1, 2x^*(u, u)$  **faça**
- 5:                  $F \leftarrow N(w) \cap N(z)$
- 6:                 Seja  $P$  os vértices de um caminho mínimo entre  $w$  e  $z$  em  $G[B \setminus F]$
- 7:                 **se**  $P \neq \emptyset$  e  $x^*(u, \langle w, v, z \rangle) > 1, 5x^*(u, u)(|P| - 2)/|P|$  **então**
- 8:                      $H \leftarrow P \cup \{v\}$  //observe que  $H$  induz um buraco em  $G$
- 9:                 **se**  $|H|$  for ímpar e  $x^*(u, H) > \frac{h-1}{2} x^*(u, u)$  **então**
- 10:                     Adicionar  $x^*(u, H) \leq \frac{h-1}{2} x^*(u, u)$  ao LP núcleo

De acordo com as propriedades 19 e 20, podemos restringir a busca por inequações violadas a vértices  $u \in V$  com  $x^*(u, v) > 0, 4x^*(u, u)$ , para algum  $v \in \bar{N}_{1/2}^+(u)$ , como feito no Algoritmo 4. Para encontrar buracos ímpares cuja inequação correspondente é violada, procuramos nas componentes conexas  $G[B]$  da anti-vizinhança reduzida de  $u$ . Esta busca

enumera caminhos da forma  $P_3 = \langle w, v, z \rangle$ , como indicado na Propriedade 20. Uma busca em largura modificada é utilizada para gerar este caminho. Esta busca consiste em ignorar os vértices de  $F$  ao fazer o percurso em  $G[B]$  para obter um caminho em  $G[B \setminus F]$ .

#### 5.4. Algoritmo de Planos de Corte

A forma geral do nosso algoritmo de planos de corte é apresentada no Algoritmo 5. Ele começa com uma fase de pré-processamento baseada nas idéias da Seção 5.1. O algoritmo de planos de corte trabalha sobre cada grafo decomposto  $G' \subseteq G$ . Utilizamos  $\mathcal{R}$ ,  $CUT$  e  $ACT$  para denotar respectivamente os conjuntos de restrições relacionados ao repositório, cortes violados e restrições que permanecem no PL núcleo pela política de manutenção de restrições. Estes conceitos são relacionados ao que é apresentado na Seção 5.2.2.

O primeiro passo ao tratar  $G'$  é fornecer uma ordem sobre os vértices e definir o modelo inicial. Estas operações são descritas na Seção 5.2.1 e na linha 4 do Algoritmo 5. Em seguida, é obtida uma solução ótima para a relaxação deste modelo inicial. A partir deste ponto, o tratamento de  $G'$  procede iterativamente, representado pelas linhas 10-23. Em cada iteração, cortes são obtidos a partir de heurísticas propostas na Seção 5.3 e do repositório e são armazenadas no conjunto  $CUT$ . Dentre as inequações previamente adicionadas ao modelo por heurísticas ou por inequações no repositório, representados pelo conjunto  $IN$  na linha 18, são removidas do modelo as inequações inativas e adicionadas ao repositório, definidas na Seção 5.2.2 e apresentados nas linhas 18, 20 e 21. Caso alguma inequação violada seja adicionada ao modelo, as iterações poderão continuar. Estas verificações são feitas na linha 10 do Algoritmo 5.

As iterações de tratamento do modelo linear seguirão até que algum ponto de parada seja obtido. Os critérios de parada incluem a solução obtida ser solução viável para o problema inteiro, como definido no Capítulo 4, nenhuma inequação violada ser adicionada ao modelo ou os limites inferiores não aumentarem rapidamente, representados pelas linhas 7, 10 e 17 deste algoritmo. Para definir o que queremos dizer pelo limite inferior não aumentar rapidamente, utilizamos os termos *tolerância* e *margem de lucro*, representados



**Algoritmo 5.** *Algoritmo de planos de corte*

- 1: Usar Algoritmo 2 sobre  $G$  para obter uma decomposição  $\mathcal{G} = \{G_1, \dots, G_j\}$
- 2:  $\mathcal{R} \leftarrow \emptyset$ ;  $c \leftarrow$  vetor de coeficientes de custo
- 3: **para todo**  $G' \in \mathcal{G}$  **faça**
- 4:   Obter um modelo inicial  $\pi(G')$
- 5:    $x^* \leftarrow$  solução ótima de  $\pi(G')$
- 6:    $bad \leftarrow 0$
- 7:   **se**  $x^*$  não é solução viável inteira **então**
- 8:      $mark \leftarrow c^T x^*$ ;  $ACT \leftarrow \emptyset$
- 9:      $CUT \leftarrow$  conjunto de cortes de  $x^*$
- 10:   **enquanto**  $bad < tol$ ,  $CUT \neq \emptyset$  e  $x^*$  não é solução viável inteira **faça**
- 11:      $x^* \leftarrow$  solução ótima de  $\pi(G') \cup ACT \cup CUT$
- 12:      $obj \leftarrow c^T x^*$
- 13:     **se**  $obj > (1 + gap)mark$  **então**
- 14:        $mark \leftarrow obj$ ;  $bad \leftarrow 0$
- 15:     **senão**
- 16:        $bad \leftarrow bad + 1$
- 17:     **se**  $bad < tol$  e  $x^*$  não é solução viável inteira **então**
- 18:        $IN \leftarrow ACT \cup CUT$
- 19:        $OUT \leftarrow \{r \in \mathcal{R} \mid r \text{ é violada por } x^*\}$
- 20:        $ACT \leftarrow \{r \in IN \mid \text{a variável dual correspondente a } r \text{ é básica}\}$
- 21:        $\mathcal{R} \leftarrow (\mathcal{R} \setminus OUT) \cup (IN \setminus ACT)$
- 22:        $CUT \leftarrow$  conjunto de cortes de  $x^*$
- 23:        $CUT \leftarrow CUT \cup OUT$

no Algoritmo 5 pelas variáveis *tol* e *gap*, respectivamente. A *marca*, representada pela variável *mark*, inicialmente é igual ao valor da função objetivo sobre o modelo inicial. Sempre que uma iteração for uma *iteração boa*, a marca é atualizada para o valor da função objetivo desta iteração. Uma iteração é uma *iteração boa* quando o valor da sua função objetivo for maior do que a marca em pelo menos o fator da *margem de lucro*. Quando uma iteração não for uma iteração boa, esta é uma *iteração ruim*. A *tolerância* é a quantidade máxima de iterações ruins sem que seja finalizado o processo de refinamento do modelo linear por cortes. Consideramos que o limite inferior não aumentou rapidamente quando o algoritmo iterar em soluções do modelo linear por uma quantidade de iterações ruins igual à tolerância definida desde a última marca.

## Experimentos e Resultados

---

Durante a elaboração desta dissertação foi implementado o algoritmo de planos de corte descrito no Capítulo 5. Segundo as heurísticas de separação implementadas, descritas na Seção 5.3, esta implementação visa resolver o desafio de aproximar  $\chi_F(G)$  de forma mais eficiente do que é conhecido atualmente, como apresentado na Seção 3.2.3. O objetivo dos nossos experimentos é determinar a robustez do limite inferior fornecido tanto quanto o tempo necessário para calcular tal limite inferior. Na Seção 6.1 descrevemos brevemente as classes de grafos que compõem o conjunto de instâncias de teste. Na Seção 6.2, discutimos alguns detalhes da nossa implementação. Na Seção 6.3 exibimos os resultados computacionais e os relacionamos com o número cromático fracionário, gerado por uma implementação da formulação de conjuntos independentes, apresentada na Seção 3.2.

### 6.1. Descrição de Instâncias

Em nossos experimentos computacionais foram utilizadas instâncias extraídas de um conjunto de problemas de coloração de grafos de DIMACS [29, 30]. Aqui, descrevemos estas classes de grafos.

**Grafos de Registros** Uma aplicação muito utilizada para coloração de grafos é o problema de alocação de registros compartilhados por variáveis de um programa, dado

um código seqüencial. Os vértices, destes grafos, representam variáveis e dois vértices são conectados por uma aresta se, e somente se, as duas variáveis correspondentes a estes vértices utilizam o mesmo registro ao mesmo tempo em um segmento de código. O número cromático destes grafos indica o número mínimo de registros necessários naquele fragmento de código. **mul soli.x** e **zero ini.x** são grafos de registros. Condon [14] desenvolveu um programa que pega fragmentos de códigos e gera o grafo correspondente.

**Grafos Rainha** Dado um tabuleiro de xadrez  $q$  por  $r$ , um grafo rainha **queen $q.r$**  é um grafo com  $qr$  vértices, cada um correspondendo a um quadrado do tabuleiro com  $q$  linhas e  $r$  colunas, e uma aresta existe entre dois vértices se, e somente se, os quadrados correspondentes a estes vértices estão em uma mesma linha, coluna ou diagonal.

**Grafos de Mycielski** Dado um grafo  $G = (V, E)$ , com  $n = |V|$ , definimos a transformação de Mycielski [38] como o processo de obtenção do grafo  $MY_G$  a partir de  $G$  onde

$$V(MY_G) = \{x_1, \dots, x_n\} \cup \{y_1, \dots, y_n\} \cup \{z\}.$$

O conjunto de arestas  $E(MY_G)$  é formado pelas arestas  $x_i x_j$  e  $x_i y_j$ , para todo  $v_i v_j \in E$ , e  $y_i z$ , para todo  $i$ . Sabemos que esta transformação não afeta o tamanho da maior clique do grafo e incrementa o número cromático em uma unidade [35]. Definimos grafos de Mycielski, denotados por **myciel $x$** , como o conjunto de  $x - 1$  transformações de Mycielski sucessivas a partir do grafo com dois vértices e uma aresta os ligando. Esta classe de grafos parece ser difícil de resolver pois não possui triângulos (a clique máxima é de cardinalidade 2), mas possui número cromático igual a  $x + 1$ .

**Grafos de Livro** Dado um trabalho de literatura, um grafo é criado onde cada vértice representa um personagem. Dois vértices são adjacentes se os personagens correspondentes se encontram no texto. Knuth [32] criou os grafos para alguns trabalhos

clássicos: *Anna Karenina* de Tolstoy (**anna**), *David Copperfield* de Dickens (**david**), *Huckleberry Finn* de Twain (**huck**) e *Les Misérables* de Hugo (**jean**).

**Grafos de Campeonato** Um grafo dos jogos de um campeonato é um grafo onde os vértices representam os participantes. Dois vértices são conectados por uma aresta se os participantes correspondentes jogaram um contra o outro no campeonato. Knuth [32] fornece o grafo do campeonato de 1990 de futebol americano entre universidades americanas, denotado por **games120**.

**Grafos de Milhas** Dado um conjunto de cidades, um grafo de  $x$  milhas destas cidades é um grafo com um vértice para cada cidade. Dois vértices são conectados por uma aresta se as cidades correspondentes estão a uma distância de no máximo  $x$  milhas uma da outra. **miles $x$**  são grafos de milhas baseados em um conjunto de cidades americanas com distância de no máximo  $x$  milhas, dadas por milhas de estradas em 1947. Estes grafos também foram fornecidos por Knuth [32].

**Grafos Aleatórios**  $\mathcal{G}_{n,p}$  são grafos com  $n$  vértices onde há uma probabilidade  $p$  de existir uma aresta entre quaisquer par de vértices, independente da existência ou não de outra aresta. **DSJCN.p** são grafos aleatórios usados em um trabalho de Johnson, Aragon, McGeoch e Schevon [28].

**Grafos flat** Os grafos **flat $n.x$**  são grafos quase-aleatórios com  $n$  vértices fornecidos por Culberson para integrar as instâncias DIMACS [30].

**Grafo mugg** O grafo **mugg88.1** foi obtido das instâncias em [29] sem descrição.

**Grafos Myciel Generalizados** Os grafos **fullins $x.y$**  e **insertions $x.y$**  são obtidos a partir de grafos de Mycielski com a inserção de vértices para aumentar o tamanho do grafo mas não a sua densidade.

## 6.2. Sumário de Implementação

Os testes foram implementados em Java utilizando um computador Pentium IV 1,4GHz com 1024MBytes de memória usando CPLEX 7.5 para resolver os problemas de programação linear. O tamanho máximo do pool de alocação de memória utilizado na Java Virtual Machine foi colocado em 512MBytes. Os resultados fornecidos tiveram tempo limite de 10 minutos. Os valores atribuídos a *tol* e *gap*, descritos na Seção 5.4, são baseados em testes realizados e serão de 5 e 0.01, respectivamente.

A implementação utiliza o pacote *pargoworks* e interfaces próprias para pacotes de resolução de problemas de programação linear, também chamados de *pacotes de PL*, cujas documentações podem ser encontradas em [10]. Como contribuição pessoal, destaco grande parte da implementação do pacote *pargoworks* e o desenvolvimento da interface para pacotes de PL. Esta interface para pacotes de PL visa uma abstração de qual pacote de PL foi utilizado. Assim, fica mais fácil a mudança de um pacote de PL para outro simplesmente ao reimplementar a interface. O pacote *pargoworks* fornece implementações de *frameworks* de desenvolvimento de algumas aplicações. Dentre elas, posso citar a utilização de grafos, *frameworks* para a solução de métodos de programação inteira, incluindo o métodos de planos de corte e o método *branch-and-bound*, e classes base para acelerar códigos feitos em Java. O método de aceleração do código em Java inclui repositórios de objetos que são necessitados com frequência. Ao reutilizar um objeto, previamente alocado mas não mais necessário, podemos acelerar os resultados obtidos ao evitar etapas do coletor de lixo do Java. Os pacotes desenvolvidos e utilizados visam uma maior produtividade de implementações, em Java, de problemas de programação inteira ou a utilização de grafos.

Alguns pontos críticos foram identificados como as operações mais custosas em termos de tempo. São eles a resolução do modelo linear e a geração de cliques maximais ponderadas. Para tirar este gargalo de tempo de processamento, foram utilizadas implementações em linguagem C++ para a resolução destes problemas. No caso da resolução do modelo linear, utilizamos a interface Java do CPLEX 7.5. Na geração de boas cliques maximais

ponderadas, utilizamos JNI, a interface nativa de Java, para encapsular a implementação da heurística QUALEX-MS.

Apresentamos uma descrição das instâncias nas tabelas 6.1–6.4. Para cada grafo  $G$ , são apresentadas as seguintes informações:

- Tamanho: informação relativa ao número de vértices  $n$  e ao número de arestas  $m$  da instância.
- CLIQUE: representa o tamanho da maior clique de encontrada em  $G$ , sendo utilizada como limite inferior pelas heurísticas de pré-processamento, descritas na Seção 5.1.
- UB: informação relativa ao melhor limite superior conhecido para a instância. Caso este limite superior seja conhecido como ótimo, este será marcado com um asterisco. Caso nenhum limite superior seja conhecido, esta coluna ficará sem valor neste campo.
- Pré-processamento: a quantidade de grafos  $|\mathcal{G}|$  obtidas na decomposição utilizada no Algoritmo 5 e o número de vértices  $n'$  de todos os grafos desta decomposição.

### 6.3. Sumário de Resultados

Os resultados dos limites inferiores sobre as instâncias descritas anteriormente são apresentados nas tabelas a seguir. Para cada grafo  $G$ , são apresentadas as seguintes informações:

- $\chi_F(G)_{\text{up}}$ : Limite superior para o número cromático fracionário. Caso seja conhecido que este limite iguala a  $\chi_F(G)$ , este valor será marcado por um asterisco. Falta de informação neste campo indica que não é conhecido o valor exato ou nenhum limite superior que seja melhor do que o valor fornecido no campo UB. Estes resultados foram obtidos utilizando o algoritmo proposto em [37].
- REP: informações relativas ao valor da solução fornecida pelo algoritmo. Estas informações se dividem em:

Grafo	Tamanho		CLIQUE	UB	Pré-processamento	
	$n$	$m$			$ \mathcal{G} $	$n'$
DSJC125.1	125	1472	4	5*	1	125
DSJC125.5	125	7782	10	18	1	125
DSJC125.9	125	13922	34	44*	1	125
DSJC250.1	250	6436	4	8	1	250
DSJC250.5	250	31336	11	29	1	250
DSJC250.9	250	55794	42	75	1	250
flat300.20	300	21375	11	20*	1	300
flat300.26	300	21633	11	26*	1	300
flat300.28	300	21695	12	28*	1	300

**Tabela 6.1.** Sumário de instâncias de grafos aleatórios ou quase-aleatórios

- LP: fornece as informações sobre o valor da função objetivo fornecido ao terminar o algoritmo. A falta de um valor neste campo indica que o modelo linear não foi resolvido dentro do tempo limite.
- Int: o limite inferior obtido ao pegar o topo do valor de função objetivo, ou seja,  $\lceil LB \rceil$ .
- T: o tempo total de execução, fornecido em segundos.
- %T(LP): a porcentagem do tempo do algoritmo que passou resolvendo o modelo linear.
- Iter: quantidade de vezes em que o modelo linear foi resolvido para obter o resultado.

Os resultados obtidos podem ser vistos nas tabelas 6.5–6.7. O limite inferior fornecido por estes resultados também é um limite inferior para o número cromático fracionário.



Grafo	Tamanho		CLIQUE	UB	Pré-processamento	
	$n$	$m$			$ \mathcal{G} $	$n'$
multisoli.1	197	3925	49	49*	0	0
multisoli.2	188	3885	31	31*	1	88
multisoli.3	184	3916	31	31*	1	88
multisoli.4	185	3946	31	31*	1	90
multisoli.5	186	3973	31	31*	1	89
zeroini.1	211	4100	49	49*	0	0
zeroini.2	211	3541	30	30*	1	45
zeroini.3	206	3540	30	30*	1	44
miles250	128	774	8	8*	1	14
miles500	128	2340	20	20*	0	0
miles750	128	4226	31	31*	0	0
miles1000	128	6432	42	42*	1	14
miles1500	128	10396	73	73*	0	0
anna	138	986	11	11*	0	0
david	87	812	11	11*	0	0
huck	74	602	11	11*	0	0
jean	80	508	10	10*	0	0
mugg88.1	88	146	3	4*	1	88
games120	120	1276	9	9*	1	117

**Tabela 6.2.** *Sumário de instâncias de grafos de registros e outros grafos diversos*

Grafo	Tamanho		CLIQUE	UB	Pré-processamento	
	$n$	$m$			$ \mathcal{G} $	$n'$
queen5.5	25	320	5	5*	1	25
queen6.6	36	580	6	7*	1	36
queen7.7	49	952	7	7*	1	49
queen8.8	64	1456	8	9*	1	64
queen8.12	96	2736	12	12*	1	96
queen9.9	81	2112	9	10*	1	81
queen10.10	100	2940	10	-	1	100
queen11.11	121	3960	11	11*	1	121
queen12.12	144	5192	12	-	1	144
queen13.13	169	6656	13	13*	1	169
queen14.14	196	8372	14	-	1	196
queen15.15	225	10360	15	-	1	225
queen16.16	256	12640	16	-	1	256

**Tabela 6.3.** *Sumário de instâncias de grafos queen*

Grafo	Tamanho		CLIQUE	UB	Pré-processamento	
	$n$	$m$			$ \mathcal{G} $	$n'$
myciel3	11	20	2	4*	1	11
myciel4	23	71	2	5*	1	23
myciel5	47	236	2	6*	1	47
myciel6	95	755	2	7*	1	95
myciel7	191	2360	2	8*	1	191
fullins1.3	30	100	3	4*	1	15
fullins1.4	93	593	3	5	1	35
fullins1.5	282	3247	3	6	1	75
fullins2.3	52	201	4	5*	1	17
fullins2.4	212	1621	4	6	1	41
fullins3.3	80	346	5	6*	1	15
fullins4.3	114	541	6	7*	1	18
fullins5.3	154	792	7	8*	1	21
insertions1.4	67	232	2	4*	1	67
insertions1.5	202	1227	2	6	1	202
insertions2.3	37	72	2	4*	1	37
insertions2.4	149	541	2	4*	1	149
insertions3.3	56	110	2	4*	1	56
insertions3.4	281	1046	2	5	1	281
insertions4.3	79	156	2	3*	1	79

**Tabela 6.4.** *Sumário de instâncias de grafos mycielski e mycielski generalizados*

Apresentamos somente os resultados relativos a instâncias cujo pré-processamento não reduziu o grafo completamente.

Grafo	$\chi_F(G)_{\text{up}}$	REP				
		LP	Int	T	%T(LP)	Iter
DSJC125.1	4,86	4,33	5	297,3	80,6	17
DSJC125.5	15,96	13,15	14	558,9	60,0	23
DSJC125.9	42,73	42,71	43	3,0	10,7	8
DSJC250.1	7,94	4,00	4	600,0	93,1	1
DSJC250.5	-	11,00	11	600,0	95,4	1
DSJC250.9	-	70,00	70	45,2	53,2	12
flat300.20	-	11,00	11	600,0	90,5	1
flat300.26	-	11,00	11	600,0	90,7	1
flat300.28	-	12,00	12	600,0	90,0	1

**Tabela 6.5.** *Sumário de resultados sobre grafos aleatórios ou quase-aleatórios*

Para observar o comportamento das heurísticas de separação, apresentamos resultados relacionados aos cortes gerados pelas heurísticas de separação. Para cada grafo  $G$ , são apresentadas as seguintes informações:

- LPI: fornece as informações sobre o valor da função objetivo alcançado pelo modelo inicial.
- LP: fornece as informações sobre o valor da função objetivo atingida ao terminar o algoritmo.
- Heurísticas de separação: fornece informações sobre os cortes gerados para a geração de cliques externas (clique) e buracos externos (buracos). As informações fornecidas são:

Grafo	$\chi_F(G)_{up}$	REP				
		LP	Int	T	%T(LP)	Iter
multisoli.2	31,00*	31,00	31	2,6	6,3	1
multisoli.3	31,00*	31,00	31	2,7	7,9	1
multisoli.4	31,00*	31,00	31	3,4	6,6	1
multisoli.5	31,00*	31,00	31	3,3	6,4	1
zeroini.2	30,00*	30,00	30	0,6	2,8	2
zeroini.3	30,00*	30,00	30	0,6	2,8	2
queen5.5	5,00*	5,00	5	0,7	2,3	3
queen6.6	7,00*	6,21	7	3,8	9,6	13
queen7.7	7,00*	7,00	7	3,8	25,4	6
queen8.8	8,44*	8,00	8	6,5	43,1	6
queen8.12	12,00*	12,00	12	8,2	41,4	6
queen9.9	9,00*	9,00	9	13,1	60,0	6
queen10.10	-	10,00	10	35,8	74,2	6
queen11.11	11,00*	11,00	11	87,7	84,6	6
queen12.12	-	12,00	12	208,3	90,7	6
queen13.13	13,00*	13,00	13	600,0	94,7	5
queen14.14	-	14,00	14	600,0	94,7	3
queen15.15	-	15,00	15	600,0	93,3	2
queen16.16	-	16,00	16	600,0	94,7	1
miles250	8,00*	8,00	8	1,2	0,5	1
miles1000	42,00*	42,00	42	0,2	3,4	1
mugg88.1	-	3,00	3	5,8	40,5	6
games120	9,00*	9,00	9	8,9	15,6	6

**Tabela 6.6.** *Sumário de resultados sobre grafos de registros, queen e outros grafos diversos*

Grafo	$\chi_F(G)_{up}$	REP				
		LP	Int	T	%T(LP)	Iter
myciel3	2,90*	2,90	3	0,6	1,6	3
myciel4	3,24*	2,91	3	1,1	4,7	6
myciel5	3,55*	3,08	4	9,2	24,5	9
myciel6	3,86	2,99	3	86,0	75,4	8
myciel7	4,25	2,63	3	600,0	87,7	1
fullins1.3	3,33*	3,33	4	0,9	6,7	5
fullins1.4	3,63	3,40	4	2,3	10,3	10
fullins1.5	4,02	3,40	4	22,2	52,5	11
fullins2.3	4,25*	4,25	5	0,7	2,1	4
fullins2.4	4,56	4,26	5	2,3	17,4	6
fullins3.3	5,20*	5,20	6	0,6	1,7	2
fullins4.3	6,22	6,17	7	0,4	2,6	1
fullins5.3	7,20	7,14	8	0,4	3,0	1
insertions1.4	2,84	2,52	3	20,1	57,9	9
insertions1.5	3,32	2,33	3	600,0	96,1	1
insertions2.3	-	2,34	3	2,5	16,8	10
insertions2.4	2,79	2,32	3	600,0	92,7	6
insertions3.3	-	2,23	3	5,7	45,7	8
insertions3.4	2,80	-	-	600,0	92,7	0
insertions4.3	2,38	2,18	3	13,7	66,8	8

**Tabela 6.7.** Sumário de resultados sobre grafos mycielski e mycielski generalizados

- Qtd: quantidade de cortes encontrados dentro do limite de tempo de processamento
- T: o tempo de execução, fornecido em segundos, necessários para gerar os cortes dentro do limite de tempo de processamento

Os cortes gerados para as respectivas instâncias podem ser vistos nas tabelas 6.8–6.10. Foram apresentados apenas os resultados referentes a instâncias que resolveram pelo menos uma vez o PL núcleo com a inclusão de cortes gerados por heurísticas de separação.

Grafo	REP		cliques		buracos	
	LPI	LP	Qtd	T	Qtd	T
DSJC125.1	4,00	4,33	3142	9,0	703	6,9
DSJC125.5	10,00	13,15	13223	44,7	3969	50,4
DSJC125.9	40,99	42,71	175	0,8	14	0,4
DSJC250.9	58,67	70,00	2201	6,3	246	3,0

**Tabela 6.8.** *Sumário de cortes gerados sobre grafos aleatórios ou quase-aleatórios*

Em uma grande quantidade das instâncias, a distância entre  $\omega(G)$  e um limite superior conhecido é muito pequena e o limite inferior fornecido pela formulação iguala um limitante superior. As instâncias em que esta formulação mostra a sua maior força ocorre na classe DSJCN.x. Estas instâncias possuem uma diferença considerável entre a maior clique e o menor limitante superior conhecido. Mesmo assim, o limite inferior consegue chegar bem próximo deste limitante superior, mesmo quando o modelo inicial fornece um limite inferior próximo de  $\omega(G)$ .

As instâncias de grafos de registros, grafos de milhas e grafos de livro são as mais fáceis de resolver. Quando o algoritmo de pré-processamento não decompõe o grafo por completo, no máximo duas iterações do modelo linear geram uma solução ótima inteira.

Grafo	REP		cliques		buracos	
	LPI	LP	Qtd	T	Qtd	T
zeroini.2	30,00	30,00	1	<0,1	0	<0,1
zeroini.3	30,00	30,00	1	<0,1	0	<0,1
queen5.5	5,00	5,00	25	0,1	0	<0,1
queen6.6	6,00	6,21	306	1,0	369	0,7
queen7.7	7,00	7,00	269	0,6	368	0,6
queen8.8	8,00	8,00	320	0,9	349	0,7
queen8.12	12,00	12,00	271	1,5	114	0,3
queen9.9	9,00	9,00	418	1,5	256	0,9
queen10.10	10,00	10,00	675	2,3	462	1,7
queen11.11	11,00	11,00	789	3,8	352	1,9
queen12.12	12,00	12,00	956	6,0	300	2,4
queen13.13	13,00	13,00	1088	9,7	323	4,6
queen14.14	14,00	14,00	1004	7,8	218	3,0
queen15.15	15,00	15,00	877	7,6	440	2,4
mugg88.1	3,00	3,00	53	0,4	20	0,3
games120	9,00	9,00	60	1,4	0	0,5

**Tabela 6.9.** *Sumário de cortes gerados sobre grafos de registros, queen e outros grafos diversos*



Grafo	REP		cliques		buracos	
	LPI	LP	Qtd	T	Qtd	T
myciel3	2,75	2,90	6	0,1	14	0,1
myciel4	2,78	2,91	86	0,2	0	0,1
myciel5	2,71	3,08	782	0,9	1203	1,3
myciel6	2,69	2,99	3159	3,0	30233	1,9
fullins1.3	3,33	3,33	4	0,1	0	<0,1
fullins1.4	3,13	3,40	208	0,4	116	0,3
fullins1.5	3,26	3,40	1238	1,9	454	1,4
fullins2.3	4,25	4,25	7	0,1	0	<0,1
fullins2.4	2,25	4,26	98	0,4	171	0,3
fullins3.3	5,20	5,20	1	0,1	0	<0,1
insertions1.4	2,30	2,52	1662	1,4	170	1,0
insertions2.3	2,25	2,34	223	0,4	35	0,5
insertions2.4	2,25	2,32	2742	6,2	0	4,8
insertions3.3	2,00	2,23	288	0,5	0	0,4
insertions4.3	2,00	2,18	266	0,8	0	0,9

**Tabela 6.10.** *Sumário de cortes gerados sobre grafos mycielski e mycielski generalizados*

Os grafos queen apresentam normalmente uma distância pequena entre um limite superior conhecido e o valor da maior clique encontrada. Quando estes valores se igualam, a instância é de fácil solução. As instâncias conhecidas para o qual estes valores são diferentes são as instâncias queen6.6 e queen8.8. Somente na instância queen6.6 o modelo conseguiu um limite inferior melhor do que o valor da clique máxima. Os grafos myciel são os grafos que apresentam a maior distância entre o limite inferior conhecido e o valor da coloração ótima. Isto decorre do fato de que esta distância também é grande para o número cromático fracionário.

A partir destes resultados, podemos verificar a qualidade do limite inferior fornecido pela implementação para  $\chi(G)$ , obtido no campo Int das tabelas. Para isto, basta compararmos este valor com  $\lceil \chi_F(G) \rceil$ . Nos casos em que o valor exato de  $\chi_F(G)$  é conhecido, não obtivemos um limite inferior que difere em mais de uma unidade de  $\lceil \chi_F(G) \rceil$ . Para os casos em que o valor exato do número cromático fracionário não é conhecido, observando o seu limite superior podemos observar que só é possível que esta diferença seja maior do que uma unidade para poucas instâncias, comprovando a força da formulação para um tempo de processamento relativamente curto.

## Conclusões

---

Durante a elaboração desta dissertação tratamos o problema de coloração de vértices por formulações de programação inteira 0-1 e implementamos a formulação dos representantes para fornecer limites inferiores para o número cromático, verificando a qualidade do limite inferior gerado comparando-o com o número cromático fracionário.

Estudamos algumas formulações de programação inteira 0-1 para o problema de coloração. Verificamos que a formulação dos representantes apresentava bons indicadores para uma boa implementação por alguns parâmetros abordados. Observamos a existência de simetria nesta formulação e a revisamos para uma versão com menos simetria. Mostramos que a maioria das facetas conhecidas anteriormente se mantinha como facetas na formulação adaptada. Também mostramos como relaxar a classe de inequações externas para englobar subestruturas para as quais esta não definia faceta. Este trabalho também apresenta uma nova demonstração para a classe de inequações interna definir faceta. Esta demonstração pode ser utilizada na formulação sem orientação para provar que esta define faceta para buracos ímpares e anti-buracos ímpares. Esta se tornou necessária pois a demonstração conhecida não foi viável para provar os mesmos resultados na formulação adaptada.

Apresentamos uma classe de inequações válidas de conjuntos independentes. A partir desta classe de inequações, é possível observar a existência de uma grande quantidade de

facetas do poliedro gerado pela formulação, dado que as condições para esta inequação definir faceta são fáceis de acontecer. Também observamos que podemos aplicar o método de planos de corte à formulação dos representantes para gerar limites inferiores tanto para  $\chi(G)$  como para  $\chi_F(G)$ , dependendo de quais classes de facetas incluímos no modelo.

Durante a elaboração da dissertação foi desenvolvido um algoritmo de planos de corte utilizando a formulação dos representantes para aproximar o número cromático fracionário. Verificamos que a formulação dos representantes se apresenta como uma alternativa viável para a formulação de conjuntos independentes maximais. Obtivemos resultados muito próximos de  $\chi_F(G)$ , o valor obtido pela relaxação da formulação CIM. Ao considerar o limite inferior gerado para o problema de coloração, podemos considerar a comparação a partir do topo destes valores. Utilizando esta idéia, não geramos um limite inferior que seja menor do que o gerado por  $\lceil \chi_F(G) \rceil$  em mais do que uma unidade, indicando a força do método.

Em termos do tempo de processamento reportado, pode-se considerar a linguagem de programação que foi utilizada. Java não é considerada uma linguagem que produziria os melhores resultados de tempo de processamento. Porém, é possível diminuir consideravelmente o tempo de processamento de uma implementação em Java utilizando algumas estratégias que utilizamos na nossa implementação. Constatamos que programas com uso intenso de memória gastam uma parte considerável do seu tempo removendo instâncias de objetos não mais utilizados com o coletor de lixo Java. Observamos que o tempo de processamento pode ser reduzido consideravelmente com reutilização de instâncias de objetos previamente alocados.

Uma direção para trabalhos futuros inclui a implementação de um algoritmo *branch-and-cut* para o problema de coloração de vértices com a geração de um limite inferior para o número cromático fracionário. Pela qualidade do limite inferior para o número cromático fracionário, podemos observar o trabalho de Mehrotra e Trick [37] para ter uma indicação de bons resultados. Outra direção inclui tentar obter limites inferiores melhores do que  $\chi_F(G)$  utilizando as facetas interna e de conjuntos independentes.

# Demonstrações das Facetas dos Representantes

---

Apresentamos as demonstrações referentes às facetas apresentadas na Seção 4.2. Estes resultados se apresentam em Inglês e são referentes ao texto submetido a publicação em [7].

## A.1. Trecho do Artigo [7] com Demonstrações

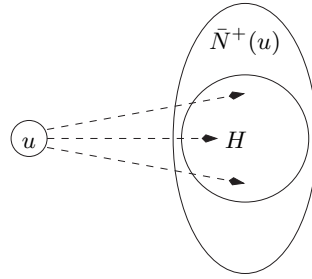
### External facets

The facet defining inequalities described in the following theorem involve a subset  $H$  of  $V$  and the maximum number of colors in  $H$  that can be represented by a vertex  $u$  outside  $H$ , as depicted in Figure A.1. We need some additional notation in this section. Call an *ordered pair*  $uv$  of  $G$  two vertices  $v \in V \setminus S$  and  $u \in \bar{N}^-[z]$ . Given an ordered pair  $uv$ , its incidence vector  $e^{uv} \in \{0, 1\}^{n+\bar{m}-|S|}$  is a vector whose elements are indexed by the ordered pairs of  $G$  in which the element indexed by  $uv$  is the only non-zero element. In addition, some special terms associated with  $H$  are used. If  $H \subseteq \bar{N}^+(u)$ , then  $X(u, H)$  stands for the  $n$ -coloring  $\sum_{v \in V \setminus S} e^{vv} + \sum_{v \in H} e^{uv}$ . For  $v \in H$ , denote by  $\alpha_v \in \mathbb{N}$  the maximum size of an independent set, of  $G[H]$ , that contains  $v$ . The maximum of these values, for all  $v \in H$ , is denoted by  $\alpha_H$ . An edge  $vw$  is said to be *safe* if there exist two independent

sets  $W_v$  and  $W_w$ , both of  $G[H]$  and of size  $\alpha$ , such that

$$W_v \setminus W_w = \{v\}, W_w \setminus W_v = \{w\} \text{ and } \alpha_z = \alpha, \forall z \in W_v \cup W_w. \quad (\text{A.1})$$

Observe that  $vw$  being safe implies that  $\alpha_v = \alpha_w$  and, by transitivity, two vertices  $v'$  and  $w'$  connected by a path of safe edges have  $\alpha_{v'} = \alpha_{w'}$ . In other words,  $\alpha_v = \alpha_w$  if  $v$  and  $w$  are two vertices in a same connected component of  $G^{\text{safe}} = (H, E^{\text{safe}})$ , where  $E^{\text{safe}}$  stands for the set of safe edges of  $G[H]$ . A generalization of the independent set inequality described in [9], which also strengthens (4.2), is given next.



**Figura A.1.** *External inequality.*

**Teorema 22.** *Let  $u \in V \setminus T$  and  $H \subseteq \bar{N}^+(u)$ . The external inequality*

$$\sum_{v \in H} \frac{1}{\alpha_v} x_{uv} \leq y_u \quad (\text{A.2})$$

*is valid. Moreover, it is facet defining if the following properties hold:*

1.  $G[H]$  is  $\alpha_H$ -maximal in  $G^+(u)$ , which means that if  $\bar{N}^+(u) \supseteq H' \supset H$ , then  $\alpha_{H'} > \alpha_H$ ; and
2. Every connected component  $C$  of  $G^{\text{safe}}$  contains an independent set  $W$  such that  $|W| = \alpha_v$ , for every vertex  $v \in W_v$ .

*Proof:* Consider  $u \in V \setminus T$ ,  $H \subseteq \bar{N}^+(u)$  and an incidence vector  $x$  of a coloring of  $G$ . If  $u$  is not a representative vertex in  $x$ , then, by (4.2), the lefthand side of the external inequality

associated with  $u$ ,  $H$  and  $x$  is null, implying that the inequality is valid. Otherwise, assume that  $u$  is a representative, which means that  $y_u = 1$ . It turns out that the vertices of  $H$  represented by  $u$  form an independent set  $W$  of  $G[H]$ . Let  $\alpha = \min_{v \in W} \alpha_v$ . Then,  $|W| \leq \alpha$  and

$$\sum_{v \in H} \frac{1}{\alpha_v} x_{uv} = \sum_{v \in W} \frac{1}{\alpha_v} \leq \frac{1}{\alpha} \sum_{v \in W} 1 \leq 1$$

It follows that the external inequality is also valid when  $u$  is a representative.

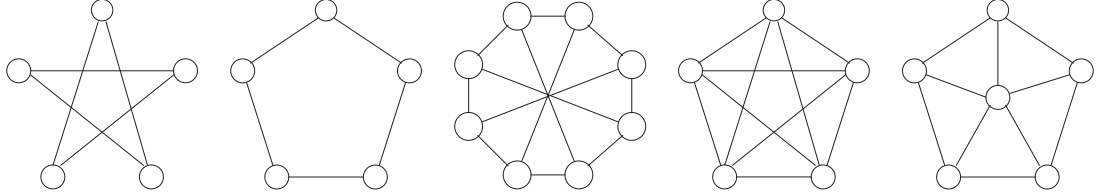
Now consider that properties 1 and 2 hold for  $u$  and  $H$ , and let us show that  $F' = \{x \in P(G) \mid \sum_{v \in H} \frac{1}{\alpha_v} x_{uv} = y_u\} \subseteq F$  defines a facet of  $P(G)$ . In what follows, we consider the entries  $\lambda_{wz}$  and  $\lambda_{zz}$  of  $\lambda$ , where  $z \in V \setminus S$  and  $w \in \bar{N}^-(z)$ .

We start showing the zero entries. Let  $W$  be a maximum independent set of  $G[H]$ . Since  $\alpha_z = |W|$ , for all  $z \in W$ , we have  $X(u, W) \in F$ . If  $w \neq u$ , the vectors  $X(u, W)$  and  $X(u, W) + e^{wz} \in F$  prove that  $\lambda_{wz} = 0$ . They also prove the same result when  $w = u$  and  $z \notin H$ , in which case we need  $W + z$  to be an independent set of  $G$ . Property 1 ensures that such a  $W$  exists. Now assume that  $z \neq u$ . If  $z \notin H$ , we have  $X(u, W) + e^{wz} \in F$ , as before. This together with  $X(u, W) + e^{wz} - e^{zz}$  show that  $\lambda_{zz} = 0$ . In the case when  $z \in H$ , let  $W_z$  be an independent set containing  $z$  such that  $\alpha_v = |W_z|$ , for all  $v \in W_z$ . Such a  $W_z$  exists either by property 2 or by condition (A.1), depending if  $z$  is isolated in  $G^{\text{safe}}$  or not. So, both  $X(u, W_z)$  and  $X(u, W_z) - e^{zz}$  are in  $F$ , which proves  $\lambda_{zz} = 0$ .

Let us turn our attention to the non-zero entries of  $\lambda$  by assuming that  $w = u$  and  $z \in H$ . Denote by  $C$  the connected component of  $G^{\text{safe}}$  that contains  $z$ . Let  $z' \in H$  and assume initially that  $zz'$  is a safe edge. Choose independent sets  $W_z$  and  $W_{z'}$  satisfying condition (A.1). Then,  $X(u, W_z)$  and  $X(u, W_{z'})$  belong to  $F$ , which implies that  $\lambda_{uz} = \lambda_{uz'}$ . This result extends to all vertices in  $C$ . Still, we need to show that the same occurs even if  $z$  and  $z'$  lie in different connected components of  $G^{\text{safe}}$ . What is required for the independent set  $W$  of  $C$  chosen in this case is that it satisfies property 2 in order that  $X(u, W) \in F$ . If  $u \in S$ , then  $\beta = \lambda(\sum_{v \in V \setminus S} e^{vv} + \sum_{v \in W} e^{uv}) = \sum_{v \in W} \lambda_{uv} = \alpha_z \lambda_{uz}$ , that is,  $\lambda_{uz} = \beta / \alpha_z$ . Otherwise,  $u \in V \setminus S$  and  $X(u, W) + e^{ru}$  and  $X(r, \{u\}) - e^{uu}$ , for any  $r \in \bar{N}^-(u)$ , belong to  $F$ . Hence, we get  $\sum_{v \in W} \lambda_{uv} = -\lambda_{uu}$ , leading to  $\lambda_{uz} = -\lambda_{uu} / \alpha_z$

and completing the proof of the theorem.  $\square$

It is worth remarking that property 2 holds for many subgraphs that are strongly related to the vertex coloring problem, namely cliques and odd holes, anti-holes and wheels (see Figure A.2).



**Figura A.2.** *Some subgraphs related to external facets.*

## Internal facets

In contrast with the inequalities of the previous section, we analyze next the inequalities derived from the minimum number of colors that should be represented by the vertices of some structure induced by a set  $H \subseteq V$ .

### Odd holes and anti-holes

The asymmetric version of the chromatic number inequality of [9] is given next. The proof presented below is considerably different from that in [9]. Let  $S_H$  be the set of minimal vertices in the sub-order induced by  $H \subseteq V$ .

**Teorema 23.** *If  $H \subseteq V$ , then the internal inequality*

$$\sum_{v \in H \setminus S_H} x((\bar{N}^-(v) \setminus H) + v, v) \geq \chi(G[H]) - |S_H| \quad (\text{A.3})$$

*is a valid inequality for  $P(G)$ . In addition, the internal inequality is facet defining if  $H$  induces an odd hole or an odd anti-hole in  $G$ .*



*Proof:* In any coloring of  $G$ , each color appearing in  $H$  that does not occur in  $S_H$  adds at least 1 to the lefthand side of the inequality. Since at least  $\chi(G[H])$  colors are necessary to color  $G[H]$ , this inequality is valid.

Let  $H \subseteq V$  be such that  $G[H]$  is an odd hole or anti-hole. Let  $F' = \{x \in P(G) \mid x \text{ satisfies (A.3) at equality}\}$ . For each case that occurs in the proof, we proceed by choosing a coloring  $\mathbf{W} = \{W_1, \dots, W_k\}$  of  $G$  using  $k \geq \chi(G)$  colors such that  $\cup_{1 \leq i \leq \chi(G[H])} W_i = H$  and, for all  $\chi(G[H]) < i \leq k$ ,  $W_i = \{v\}$ , with  $v \in V \setminus H$ . Notice that, for any such  $k$ -coloring  $\mathbf{W}$ , its incidence vector  $X^{\mathbf{W}}$  belongs to  $F'$ . Since  $G[H]$  is  $\chi(G[H])$ -critical, for each  $v \in H$  we can choose a particular of these colorings  $\mathbf{W}$ , to be denoted  $\mathbf{W}^v$ , such that  $W_1 = \{v\}$ .

To show the zero entries of  $\lambda$ , take  $v \in V \setminus S$  and  $u \in \bar{N}^-(v)$  such that  $u \in H$  or  $v \notin H \setminus S_H$ . If  $u \in H$ , choose  $X^{\mathbf{W}^u}$  and  $X^{\mathbf{W}^u} + e^{uv}$ , both in  $F$ , to prove that  $\lambda_{uv} = 0$ . In the case that  $u \notin H$  and  $v \notin H \setminus S_H$ , choose any of the colorings  $\mathbf{W}$ , or specifically  $\mathbf{W} = \mathbf{W}^v$  if  $v \in S_H$ . The vectors  $X^{\mathbf{W}}$ ,  $X^{\mathbf{W}} + e^{uv}$  and  $X^{\mathbf{W}} + e^{uv} - e^{vv}$ , all in  $F$ , prove that  $\lambda_{uv} = \lambda_{vv} = 0$ .

For the non-zero entries of  $\lambda$ , let  $v \in H \setminus S_H$  and  $u \in \bar{N}^-(v) \setminus H$ . The vectors  $X^{\mathbf{W}^v}$  and  $X^{\mathbf{W}^v} + e^{uv} - e^{vv}$ , both in  $F$ , show that  $\lambda_{uv} = \lambda_{vv}$ . To prove the remaining equalities between the non-zero entries of  $\lambda$ , we separately consider the cases where  $G[H]$  is an odd anti-hole or an odd hole. In the first case, consider  $w \in H \setminus S_H$  with  $v \prec w$  and  $vw \in \bar{E}$ . Also, take  $z \in \bar{N}^-(v) \cap H$ . Notice that  $\mathbf{W}^z$  contains the color classes  $\{z\}$  and  $\{v, w\}$ , implying that  $X^{\mathbf{W}^z}$  and  $X^{\mathbf{W}^z} - e^{vv} - e^{vw} + e^{ww} + e^{zv}$  are in  $F$ . These vectors and the fact that  $\lambda_{vv} = \lambda_{zv} = 0$  yield that  $\lambda_{vv} = \lambda_{vw}$ . Moreover, this equality extends to all vertices lying in the same connected component of  $\bar{G}[H \setminus S_H]$ . Now consider two connected components  $C_1$  and  $C_2$  of  $\bar{G}[H \setminus S_H]$  which are consecutives in the hole  $\bar{G}[H]$ . Then, there is  $s \in S_H$  connecting  $C_1$  and  $C_2$ . Let  $v \in C_1$  and  $w \in C_2$  such that  $sv, sw \in \bar{E}$ . Using a similar argument, we can use the vectors  $X^{\mathbf{W}^v}$  and  $X^{\mathbf{W}^v} - e^{vv} - e^{sw} + e^{ww} + e^{sv}$ , both in  $F$ , to show that  $\lambda_{vv} = \lambda_{vw}$ .

Now assume that  $H$  induces an odd hole. Let  $v_1 \prec v_2 \prec v_3$  be the three smallest vertices of  $H$  with respect to  $\prec$ . For each  $v \in H - v_1$ , denote by  $W_1 = \{v\}$ ,  $W_2$  and  $W_3$  the three color classes of  $\mathbf{W}^v$ , being  $W_2$  the one that contains  $v_1$ . We have that  $W_2$  contains one of the adjacent vertices of  $v$  and  $W_3$  contains the other one. Let us partition  $H \setminus \{v_1, v_2\}$  into two subsets  $H_1$  and  $H_2$  that induce an odd path and an even path, respectively. Notice that only  $H_1$  may be empty. First, we show that all  $\lambda_{vv}$  in each subset are equal. For subset  $H_2$ , we use the fact that  $v_2 \in W_3$  in the coloring  $\mathbf{W}^v$ , for any  $v \in H_2$ . Thus, for two distinct vertices  $v, w \in H_2$ , the vectors  $X^{\mathbf{W}^v}$  and  $X^{\mathbf{W}^w}$  show that  $\lambda_{vv} = \lambda_{ww}$ , since  $v_1$  and  $v_2$  always represent the color classes  $W_2$  and  $W_3$ , respectively. If  $H_1 = \emptyset$ , which means that  $v_2 \in S_H$ , the proof is complete. Otherwise, let  $v \in H_1$ . Notice that  $v_2 \in W_2$  in  $X^{\mathbf{W}^v}$ . Now we can find a coloring  $\mathbf{W}'$  that, together with  $\mathbf{W}^v$ , proves that  $\lambda_{vv} = \lambda_{v_2v_2}$ . If  $vv_2 \in E$ ,  $\mathbf{W}'$  is obtained by interchanging  $v_2$  and  $v$  in  $\mathbf{W}^v$ . Otherwise,  $\mathbf{W}'$  is derived from  $\mathbf{W}^v$  by adding  $v_2$  to  $W_1$ . If  $H_2 = \{v_3\}$ , the proof is complete as  $v_3 \in S_H$ . Otherwise, we still have to consider two cases with respect to the minimum vertex  $v$  of  $H_1$ . If there is  $w \in H_2$  such that  $v \prec w$ , the colorings  $\mathbf{W}^w$  and  $\mathbf{W}'$ , which adds  $v$  to  $W_1$  in  $\mathbf{W}^w$ , imply that  $\lambda_{vv} = \lambda_{ww}$ . In the complementary case,  $H_2$  must include  $v_3$  and two other vertices  $u, w$ , with  $u \prec w \prec v$ , which are the endpoints of the path  $G[H_1]$ . Thus, in  $\mathbf{W}^v$ , we have that  $W_1 = \{v\}$ ,  $\{v_1, v_2\} \subset W_2$  and  $\{u, w\} \subset W_3$ . Therefore,  $\mathbf{W}^v$  and the coloring obtained by transferring  $w$  to  $W_1$  show that  $\lambda_{vv} = \lambda_{ww}$ . In any case, we get an equality between the entries indexed by  $H_1$  and those indexed by  $H_2 + v_2$ . This completes the proof.  $\square$

### Independent sets

A new class of facet defining inequalities emerges from the following observation.

**Fact 1.** *Let  $H_0 \subseteq V \setminus S$  be an independent set of  $G$  with  $|H_0| \geq 2$ . Let  $v_0$  be the minimal vertex of  $H_0$  with respect to  $\prec$  and  $f(v_0)$  be any vertex in  $H_0 - v_0$ . Then, it follows from inequalities (4.1) that one of the following conditions holds for any coloring of  $G$ :*

1.  $H_0$  contains at least two representatives, or

2.  $v_0$  is a representative and represents  $f(v_0)$ , or
3.  $v_0$  or  $f(v_0)$  are represented by anti-neighbors outside  $H_0$ .

Consequently, a valid inequality is

$$\Phi(H_0) = \sum_{v \in H_0} x(v, v) + x(v_0, f(v_0)) + \sum_{v \in \{v_0, f(v_0)\}} x(\bar{N}^-(v) \setminus H_0, v) \geq 2. \quad (\text{A.4})$$

Inequality (A.4) does not define a facet of  $P(G)$ . Indeed, it is dominated by the summation of inequalities (4.1), for  $v_0$  and  $f(v_0)$ , inequalities  $x_{uf(v_0)} \leq x_{uu}$ , for all  $u \in \bar{N}^-(f(v_0)) \cap H_0 - v_0$  and  $x_{uu} \geq 0$ , for all  $u \in H_0 \setminus \bar{N}^-[f(v_0)]$ . However, the appropriate introduction of vertices in  $H_0$  strengthens the valid inequality (A.4) as follows. Let  $\{v_1, \dots, v_\ell\}$ , for an  $\ell > 0$ , be a set of vertices in  $V \setminus S$  such that the union of this set with  $H_0$  constitutes an independent set of  $G$ . By the sake of simplicity of notation, assume that  $v_i \prec v_{i-1}$ , for all  $i \in \{1, \dots, \ell\}$ . For  $H_i = H_0 \cup \{v_1, \dots, v_i\}$  and  $f(v_i) \in H_0$ , define, recursively, the following inequality:

$$\Phi(H_i) = \Phi(H_{i-1}) + x(v_i, \{v_i, f(v_i)\}) - x(v_i, v_0 + f(H_{i-1})) + \Delta_i \geq 2, \quad (\text{A.5})$$

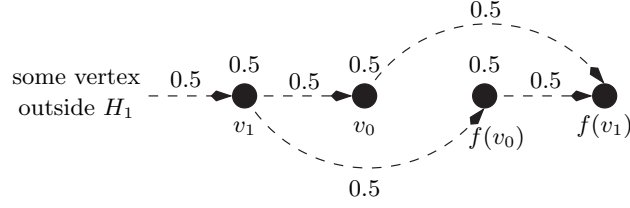
where  $f(H_{i-1}) = \cup_{j=0}^{i-1} \{f(v_j)\}$  and

$$\Delta_i = \begin{cases} x(\bar{N}^-(f(v_i)) \setminus H_i, f(v_i)), & \text{if } f(v_i) \notin f(H_{i-1}), \\ 0, & \text{otherwise.} \end{cases}$$

We can easily obtain the following equivalent nonrecursive expression:

$$\Phi(H_i) = \sum_{v \in H_i} x(v, v) + \sum_{j=0}^i x(v_j, f(v_j)) + \sum_{v \in v_0 + f(H_i)} x(\bar{N}^-(v) \setminus H_i, v) \quad (\text{A.6})$$

The next theorem points out that inequality (A.5) is facet defining in certain cases. This fact is suggested by the example in Figure A.1, which illustrates a fractional solution satisfying (4.1)-(4.2) that violates (A.5).



**Figura A.3.** Example to show that inequalities (4.1)-(4.2) do not imply  $\Phi(H_i) \geq 2$ ,  $i \in \{1, \dots, \ell\}$ , when integrality constraints are replaced by  $0 \leq x_{uv} \leq 1$ , for all  $u \in V \setminus T$  and  $v \in \bar{N}^+[u]$ . Dashed lines indicate the variables whose values are shown in the figure. The values of the variables associated with the vertices are also shown.

**Teorema 24.** The inequality  $\Phi(H_i) \geq 2$ , for all  $i \in \{0, \dots, \ell\}$ , is valid for  $P(G)$ , provided that  $H_i \cap S = \emptyset$ . In addition, it is facet defining if three conditions hold, namely

1.  $|f(H_i) - v_0| \geq 2$ ,
2. the maximum vertex of  $H_0$  with respect to  $\prec$  belongs to  $f(H_i)$ ,
3. for all  $u \in H_i \setminus \{v_0 + f(H_i)\}$ , every vertex in  $\bar{N}^+(v)$  has an anti-neighbor in  $f(H_i) \cap \bar{N}^+(u)$ , with  $v \neq f(u)$  if  $u \in H_i \setminus H_0$ .

*Proof:* To show that the inequality is valid, use induction on  $i$ . By A.6, we have  $\Phi(H_{i-1}) - x(v_i, v_0 + f(H_{i-1})) \geq 0$ . If  $\Phi(H_{i-1}) - x(v_i, v_0 + f(H_{i-1})) \geq 2$ , we are done. Thus, suppose that

$$\Phi(H_{i-1}) - x(v_i, v_0 + f(H_{i-1})) < 2.$$

Since  $\Phi(H_{i-1}) \geq 2$  by induction hypothesis, it follows that  $x(v_i, v_0 + f(H_{i-1})) \geq 1$ . Then, by inequality (4.2),  $x(v_i, v_i) = 1$ . This assures that  $\Phi(H_i) \geq 2$  if  $\Phi(H_{i-1}) - x(v_i, v_0 + f(H_{i-1})) \geq 1$  or  $\Delta_i \geq 1$ . If it is not the case, the remaining possibility is  $\Phi(H_{i-1}) - x(v_i, v_0 + f(H_{i-1})) = 0$  and  $\Delta_i = 0$ , which means that the only non-zero terms in  $\Phi(H_{i-1})$  are those in  $x(v_i, v_0 + f(H_{i-1}))$  and  $f(v_i) \in f(H_{i-1})$ . By constraints (4.1)-(4.2),  $x(v_i, f(v_i)) = 1$ , and the validity still follows.

From now on, we will be using the equivalent expression (A.6) to prove that  $\Phi(H_i) \geq 2$  defines a facet under the conditions of the theorem. We denote by  $\mathbf{W}_0$  and  $\mathbf{W}_1$  the sets of color classes  $\{\{w\} : w \notin H_i\}$  and  $\{\{s_k, v_k\} : k \in \{1, \dots, i\}\}$ , respectively, for some  $s_k \in S \cap \bar{N}^-(v_k)$ . Notice that such an  $s_k$  exists since  $H_i \subset V \setminus S$  and  $G[S]$  is a maximal clique. Still, observe that the non-zero elements of the incidence vectors of these color class do not appear in (A.6).

Let us consider the entries of  $\lambda$  indexed by  $u, v \in V$  such that  $u \in \bar{N}^-[v]$  and  $v \notin S$ . First, we exhibit a coloring  $\mathbf{W}$  such that the vectors  $X^{\mathbf{W}}$  and  $X^{\mathbf{W}} + e^{uv}$ , both satisfying  $\Phi(H_i) = 2$ , show that  $\lambda_{uv} = 0$ . We start examining the cases when  $v \notin H_i$ :

1.  $u \notin H_i, u = v$ : since  $v \notin S$ , it has an anti-neighbor in  $S$ , say  $s$ . Take the coloring  $\mathbf{W}$  whose color classes are  $\mathbf{W}_0 \setminus \{\{s\}, \{u\}\}, H_i$  and  $\{s, u\}$ .
2.  $u \notin H_i, u \neq v$ : take the coloring  $\mathbf{W}$  whose color classes are  $\mathbf{W}_0$  and  $H_i$ .
3.  $u \in H_i \setminus \{v_0 + f(H_i)\}$ : by condition 3, there is  $r \in \{1, \dots, i\}$ , such that  $f(v_r) \in \bar{N}^+(u) \cap \bar{N}(u)$ , and  $f(v_r) \neq f(u)$  if  $u \in H_i \setminus H_0$ . Besides  $\mathbf{W}_0$  and  $\mathbf{W}_1 - \{s_r, v_r\}$ , choose  $\{u, f(v_r)\}$  and  $H_0 + v_r - f(v_r)$  as the color classes of  $\mathbf{W}$ .
4.  $u \in f(H_i) - v_0$ : now the color classes of  $\mathbf{W}$  are  $\mathbf{W}_0, \mathbf{W}_1 - \{s_r, v_r\}, H_0 + v_r - u$  and  $\{u\}$ , where  $f(v_r) = u$ .
5.  $u = v_0$ : besides  $\mathbf{W}_0$  and  $\mathbf{W}_1$ , the coloring  $\mathbf{W}$  comprises the color classes  $H_0 - v_0$  and  $\{v_0\}$ .

Now, let  $v \in H_i$  and  $u \neq v$  such that:

1.  $v \notin H_0$  and  $u \notin H_i$ : the color classes of  $\mathbf{W}$  for this case are  $\mathbf{W}_0, v + H_0$  and  $\{s_k, v_k\} \in \mathbf{W}_1$ , for all  $v_k \neq v$ .
2.  $v \notin H_0$  and  $u \in H_i$ : the coloring  $\mathbf{W}$  is defined by the color classes  $\mathbf{W}_0, u + H_0$  and  $\{s_k, v_k\} \in \mathbf{W}_1$ , for all  $v_k \neq u$ .

3.  $v \in H_0 \setminus \{v_0 + f(H_i)\}$  and  $u \in H_i$ : the color classes  $\mathbf{W}_0$ ,  $\mathbf{W}_1$  and  $H_0$  form  $\mathbf{W}$  in this case.
4.  $v \in H_0$ ,  $u \in H_i$  and  $v \neq f(u)$ , if  $u \in \{v_0, \dots, v_i\}$ : let  $u' \neq u$  be a vertex in  $\{v_0, \dots, v_i\}$  such that, if  $u \in \{v_0, \dots, v_i\}$ , then  $f(u) \neq f(u')$ . Since  $f(H_i)$  has at least two vertices by condition 1, such an  $u'$  exists. The coloring  $\mathbf{W}$  is formed by the color classes  $\mathbf{W}_0$ ,  $\{u, f(u')\}$ ,  $u' + H_0 - f(u')$  and  $\{s_k, v_k\} \in \mathbf{W}_1$ , for all  $v_k \neq u, u'$ .

Let us turn our attention to the non-zero entries of  $\lambda$ . To show that all of them are equal, we start with the following claim:

**Claim 2.** *Let  $j \in \{0, 1, \dots, i\}$ . Then,  $\lambda_{v_j f(v_j)} = \lambda_{f(v_j) f(v_j)}$ . Moreover, if  $t \in \{0, 1, \dots, i\}$  and  $f(v_t) \neq f(v_j)$  then  $\lambda_{f(v_j) f(v_j)} = \lambda_{v_t v_t}$ .*

*Proof:* Consider the colorings  $\mathbf{W}_a$  and  $\mathbf{W}_b$  which are formed by the color classes  $\mathbf{W}_0$ ,  $\mathbf{W}'_1$ ,  $H_0 + v_j$ , and  $\mathbf{W}_0$ ,  $\mathbf{W}'_1$ ,  $H_0 + v_j - f(v_j)$ ,  $\{f(v_j)\}$ , respectively, where  $\mathbf{W}'_1 = \mathbf{W}_1$ , if  $j = 0$ , or  $\mathbf{W}'_1 = \mathbf{W}_1 \setminus \{s_j, v_j\}$ , otherwise. The vectors  $X^{\mathbf{W}_a}$  and  $X^{\mathbf{W}_b}$  yield the first part of the claim. In addition, the coloring  $\mathbf{W}_c$  obtained from  $\mathbf{W}_b$  by replacing the color class  $\{f(v_j)\}$  by  $\{v_t, f(v_j)\}$  defines a vector  $X^{\mathbf{W}_c}$  that, together with  $X^{\mathbf{W}_b}$ , prove the second part of the claim.  $\square$

We use this claim in the following cases:

1.  $v \in f(H_i) - v_0$ : By condition 1, there are  $j, t \in \{0, 1, \dots, i\}$  such that  $v = f(v_j) \neq f(v_t)$ . If  $v \neq f(v_0)$ , Claim 2 yields that  $\lambda_{vv} = \lambda_{v_0 v_0}$ . For  $v = f(v_0)$ , consider two cases. If  $v_0 \notin f(H_i)$ , Claim 2 together with the two colorings comprising  $\mathbf{W}_0$ ,  $\mathbf{W}_1$ ,  $H_0 - v_0$  and either  $\{v_0\}$  or  $\{v_t, v_0\}$  show that  $\lambda_{f(v_0) f(v_0)} = \lambda_{v_t v_t} = \lambda_{v_0 v_0}$ . Otherwise,  $|f(H_i)| = |H_0| \geq 3$ , which implies that there is  $r \in \{0, 1, \dots, i\}$  such that  $f(v_r) \neq f(v_0)$  and  $f(v_r) \neq f(v_t) \neq f(v_0)$ . Thus, Claim 2 proves that  $\lambda_{f(v_0) f(v_0)} = \lambda_{v_r v_r} = \lambda_{f(v_t) f(v_t)} = \lambda_{v_0 v_0}$ . In any case,  $\lambda_{vv} = \lambda_{v_0 v_0}$ .
2.  $v \in H_0 \setminus \{v_0 + f(H_i)\}$ : according to condition 2, for some  $j \in \{0, 1, \dots, i\}$ ,  $f(v_j)$  is the maximum vertex  $m$  in  $H_0$ . Using the colorings defined by  $\mathbf{W}_0$ ,  $\mathbf{W}_1$  and

either  $H_0 + v_j - m$  and  $\{m\}$  or  $H_0 + v_j \setminus \{m, v\}$  and  $\{v, m\}$  we can show that  $\lambda_{vv} = \lambda_{mm} = \lambda_{v_0v_0}$ .

3.  $v \in v_0 + f(H_i)$  and  $u \in \bar{N}^-(v) \setminus H_i$ : if  $v = v_0$ , either  $\{v_0\}$  or  $\{u, v_0\}$ , together with  $\mathbf{W}_0, \mathbf{W}_1$  and  $H_0 - v_0$ , imply that  $\lambda_{uv} = \lambda_{v_0v_0}$ . For  $v \neq v_0$ , the same equality follows by using the two colorings defined by  $\mathbf{W}_0, \mathbf{W}_1, H_0 + v_j - v$  and either  $\{v\}$  or  $\{u, v\}$ , where  $v = f(v_j)$ .
4.  $v_j \in H_i$ , for  $j \in \{0, 1, \dots, i\}$ : by Claim 2,  $\lambda_{v_j f(v_j)} = \lambda_{f(v_j) f(v_j)} = \lambda_{v_0v_0}$ . As  $|H_0 - v_0| \geq 2$ , there is  $t \in \{0, 1, \dots, i\}$  such that  $f(v_t) \neq f(v_j)$ . It follows from Claim 2 that  $\lambda_{v_j v_j} = \lambda_{f(v_t) f(v_t)} = \lambda_{v_0v_0}$ .

This shows that all non-zero entries are equal. □

We can prove that conditions 1-2 are also necessary for the inequality to define a facet. Indeed, if  $f(H_i) - v_0$  has at most one vertex, we can partition  $H_i \setminus H_0$  into two disjoint sets, namely  $A = \{v \in H_i \setminus H_0 : f(v) = v_0\}$  and  $B = \{v \in H_i \setminus H_0 : f(v) = f(v_0)\}$ . Thus, the inequality  $\Phi(H_i) \geq 2$  is given by the summation of inequalities (4.1), for  $v_0$  and  $f(v_0)$ , inequalities  $x_{uf(v_0)} \leq x_{uu}$ , for all  $u \in A$  and all  $u \in N^-(f(v_0)) \cap H_0 - v_0$ , inequalities  $x_{uv_0} \leq x_{uu}$ , for all  $u \in B$ , and  $x_{uu} \geq 0$ , for all  $u \in H_0 \setminus N^-[f(v_0)]$ . On the other hand, if the maximum vertex  $m$  of  $H_0$  does not belong to  $f(H_i)$ , the inequality is dominated by  $\Phi(H_i - m)$ .

The result established in Theorem 24 can be generalized if we consider the effect of the vertices that violate condition 3. To deal with such a situation, one should replace the inequality  $\Phi(H_i) \geq 2$  by

$$\Phi(H_i) - \sum_{u \in H_0 \setminus \{v_0 + f(H_i)\}} (u, K_u) \geq 2,$$

where  $K_u$  is a maximal clique in the set

$$Q_u = \{z \in \bar{N}^+(u) \mid N(z) \supseteq f(H_i) \cap \bar{N}^+(u) - f(v)\},$$

provided that we consider  $\{f(u)\} = \emptyset$  if  $u \in H_0$ . The proof of Theorem 24 can be slightly modified to take into account the cases forbidden by condition 3 as follows.

*Proof:* Regarding the proof of Theorem 24, we only have to consider the following additional cases:

1.  $u \in H_i \setminus \{v_0 + f(H_i)\}$ ,  $v \in Q_u \setminus K_u$ :  $\mathbf{W}_0, \mathbf{W}_1, H_0$  and either  $\{u, v'\}$  or  $\{u, v', v\}$ , for  $v \in K_u$ , which shows that  $\lambda_{uv} = 0$ . Notice that  $v$  and  $v'$  are antineighbors because  $K_u$  is maximal.
2.  $u \in H_i \setminus \{v_0 + f(H_i)\}$ ,  $v \in K_u$ :  $\mathbf{W}_0, \mathbf{W}_1, H_0$  with or without  $\{u, v\}$ , which shows that  $\lambda_{uv} = -\lambda_{uu}$ .

□



## Referências Bibliográficas

---

- [1] J. Alfonsín and B. Reed, editors. *Perfect Graphs*. Wiley Series in Discrete Mathematics and Optimization. John Wiley & Sons, 2001.
- [2] M. Bazaraa, J. Jarvis, and H. Sherali. *Linear Programming and Network Flows*. John Wiley and Sons, 1977.
- [3] B. Bollob'as and A. Thomason. Set coloring of graphs. *Discrete Mathematics*, 25:27–31, 1979.
- [4] D. Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.
- [5] J. Brown. Chromatic scheduling and the chromatic number problem. *Management Science*, 19(4):456–463, 1972.
- [6] M. Campêlo, V. Campos, and R. Corrêa. Demonstrações em anexo. Obtidas a partir de [7].
- [7] M. Campêlo, V. Campos, and R. Corrêa. On the asymmetric representatives formulation for the vertex coloring problem. Submitted for publication in a special volume of Discrete Applied Mathematics.

- [8] M. Campêlo, V. Campos, and R. Corrêa. On the asymmetric representatives formulation for the vertex coloring problem. In *Proceedings of the Brazilian Symposium on Graphs, Algorithms and Combinatorics - GRACO 2005*, volume 19 of *Electronic Notes in Discrete Mathematics*, pages 337–343, 2005.
- [9] M. Campêlo, R. Corrêa, and Y. Frota. Cliques, holes and the vertex coloring polytope. *Information Processing Letters*, 89:159–164, 2004.
- [10] V. Campos and R. Corrêa. Pargoworks documentation. <http://www.lia.ufc.br/~pargo/doc/pargoworks/index.html>.
- [11] F. Chow and J. Hennessy. Register allocation by priority-based coloring. *ACM SIGPLAN Notices*, 19(6):222–232, June 1984.
- [12] F. Chow and J. Hennessy. The priority-based coloring approach to register allocation. *ACM Transactions on Programming Languages and Systems*, 12(4):501–536, 1990.
- [13] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. Progress on perfect graphs. *Math. Program.*, Series B(97 (1–2)):405–422, 2003.
- [14] A. Condon. Dimacs challenge. Pelo desenvolvimento de um programa para gerar grafos de registros a partir de um fragmento de código, 1994.
- [15] G. B. Dantzig, D. R. Fulkerson, and S. M. Johnson. Solutions of a large-scale traveling salesman problem. *Operations Research*, 2:393–410, 1954.
- [16] D. de Werra. An introduction to timetabling. *European Journal of Operations Research*, 19:151–162, 1985.
- [17] R. Deming. Acyclic orientations of a graph and chromatic and independence numbers. *Journal of Combinatorial Theory*, Series B(26):101–110, 1979.

- [18] I. Díaz and P. Zabala. A polyhedral approach for graph coloring. *Electronic Notes in Discrete Mathematics*, 7, 2001. Brazilian Symposium on Graphs, Algorithms and Combinatorics - GRACO 2001.
- [19] C. E. Ferreira and Y. Wakabayashi. *Combinatória Poliédrica e Planos-De-Corte Faciais*. X ESCOLA DE COMPUTACAO. ed. Campinas, SP, 1996.
- [20] R. Figueiredo, V. Barbosa, N. Maculan, and C. Souza. New 0-1 integer formulations of the graph coloring problem. In *Proceedings of CLAIO 2002*, Chile, 2002.
- [21] Y. Frota. Estudo computacional de algoritmos exatos para coloração de grafos. Master's thesis, Universidade Federal do Ceará, December 2001.
- [22] A. Gamst. Some lower bounds for a class of frequency assignment problems. *IEEE Transactions on Vehicular Technology*, 35(1):8–14, 1986.
- [23] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
- [24] F. Glover, M. Parker, and J. Ryan. Coloring by tabu branch-and-bound. In Johnson and Trick [30], pages 285–307. Também encontrado em <http://mat.gsia.cmu.edu/COLOR/instances.html>.
- [25] P. Hell and F. Roberts. Analogues of the shannon capacity of a graph. *Ann. Disc. Math.*, 12:155–162, 1982.
- [26] A. Hertz. Using tabu search for graph coloring. *Computing*, 39:345–351, 1987.
- [27] A. Hilton, R. Rado, and S. Scott. A  $(j+5)$ -colour theorem for planar graphs. *Bull. London Math. Soc.*, 5:302–306, 1973.
- [28] Johnson, Aragon, McGeoch, and Schevon. Optimization by simulated annealing: An experimental evaluation; part ii, graph coloring and number partitioning. *Operations Research*, 31:378–406, 1991.

- [29] D. Johnson, A. Mehrotra, and M. Trick, editors. *Computational Series: Graph Coloring and its Generalizations*, 2004. Instâncias de coloração de grafos encontrados em <http://mat.gsia.cmu.edu/COLORING02/>.
- [30] D. Johnson and M. Trick, editors. *Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge*, volume 26 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1996. Também encontrado em <http://mat.gsia.cmu.edu/COLOR/instances.html>.
- [31] R. Karp. Reducibility among combinatorial problems. *Complexity of Computations, Advances in Computing Research*, pages 85–103, 1972.
- [32] D. Knuth. *The Stanford GraphBase*. ACM Press. Addison Wesley, New York, 1993.
- [33] B. Kubale and P. Pardalos. An algorithm for finding the maximum clique on an arbitrary graph. *DIMACS Challenge*, 1993.
- [34] M. Kubale and B. Jackowski. A generalized implicit enumeration algorithm for graph coloring. *Communications of the ACM*, 28(4):412–418, 1985.
- [35] M. Larsen, J. Propp, and D. Ullman. The fractional chromatic number of a graph and a construction of mycielski. *Journal of Graph Theory*, 19(3):411–416, 1995.
- [36] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.
- [37] A. Mehrotra and M. Trick. A column generation approach for graph coloring. *INFORMS Journal in Computing*, 8(4):344–354, 1996.
- [38] J. Mycielski. Sur le coloriage des graphes. *Colloquim Mathematiques*, 3:161–162, 1955.
- [39] G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, 1999.

- [40] J. Peemöller. A correction to brélaz's modification of brown's coloring algorithm. *Communications of the ACM*, 26(8):595–597, 1983.
- [41] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, MA, USA, 1996.
- [42] E. Sewell. An improved algorithm for exact graph coloring. *ORSA Journal on Computing*, 3:226–373, 1996.