

Limites Inferiores para o Problema de Coloração de Vértices via Geração de Cortes e Colunas

Carlos Diego Rodrigues

Mestrado e Doutorado em Ciência da Computação

Universidade Federal do Ceará - UFC

diego@lia.ufc.br

Manoel Bezerra Campêlo Neto

orientador

22 de novembro de 2005

Sumário

1	Introdução	3
2	O Problema de Coloração	5
2.1	Definição do Problema	5
2.2	Outras Definições e Notações	6
2.2.1	Teoria dos Grafos	6
2.2.2	Teoria Poliédrica	8
2.2.3	Otimização Matemática	10
2.3	Formulações de Programação Inteira	13
2.3.1	Formulação por Conjuntos Independentes Maximais (CIM)	15
2.3.2	Formulação por Vértices Representantes de Cor (VR)	16
2.4	Abordagens para o Limite Inferior	17
2.4.1	Subgrafos Críticos	18
2.4.2	Programação Matemática	20
3	Métodos de Solução	22
3.1	Geração de Colunas	22
3.1.1	Princípio da Geração de Colunas	22
3.1.2	Decomposição de Dantzig-Wolfe	24
3.2	Geração de Linhas	27
3.2.1	Princípio da Geração de Linhas	27
3.2.2	Métodos de Planos de Corte	28
3.3	Geração Simultânea de Linhas e Colunas	29
3.3.1	Princípio da Geração Simultânea	29
3.3.2	Adicionando cortes à Decomposição Dantzig-Wolfe	30

4	Formulação CIM via Formulação VR	32
4.1	Equivalência entre as Formulações	32
4.2	Desigualdades Válidas para CIM	38
4.3	Cálculo do Limite Inferior	47
5	Algoritmos	51
5.1	Algoritmos Preliminares	51
5.1.1	Algoritmo para Clique Maximal	51
5.1.2	Algoritmo para todas as cliques maximais de tamanho fixo	52
5.1.3	Algoritmo para pré-processamento	54
5.2	Algoritmos para Geração de Cortes	55
5.3	Algoritmos para Geração de Colunas	57
5.4	Algoritmo Geral	60
6	Resultados	65
6.1	Descrição das Instâncias	65
6.2	Detalhes da Implementação	66
6.3	Tabelas de Resultados	67
6.4	Análise dos resultados	73
7	Conclusões	75

Capítulo 1

Introdução

Uma k -coloração de um grafo G é a atribuição de uma dentre k cores a cada um dos vértices de G de modo que dois vértices adjacentes em G não podem receber a mesma cor. O problema de coloração de vértices pode ser definido como encontrar o menor número $\chi(G)$, denominado número cromático de G , tal que G admite uma $\chi(G)$ -coloração.

O problema de coloração de vértices é um dos mais estudados dentro da teoria dos grafos, em decorrência de uma vasta gama de aplicações teóricas e práticas. O problema encontra-se na classe NP -difícil e, portanto, a menos que $P = NP$, não existe um algoritmo polinomial para a coloração de vértices [19]. De fato, é improvável que exista até mesmo um esquema de aproximação polinomial para o problema [28].

Como já era esperado, por se tratar de um problema NP -difícil, os métodos de resolução exata para o problema se tornam inviáveis conforme aumenta-se o número de vértices do grafo. E mesmo para grafos relativamente pequenos, esses métodos extrapolam, em geral, os limites de tempo aceitáveis. Um dos elementos importantes para melhorar a eficiência desses métodos é a determinação de bons limites inferiores e superiores para o valor ótimo do problema.

Nesta dissertação procuramos encontrar melhores limites inferiores para o problema de coloração. Neste sentido, trabalharemos de duas maneiras: via subgrafos críticos e, especialmente, via programação matemática. Nesta segunda linha, vamos utilizar técnicas de geração de linhas e geração de colunas para resolver uma formulação do problema descrita por Mehrotra e Trick [29] chamada de formulação por conjuntos independentes. Vamos também utilizar, especialmente, os resultados de Campêlo, Corrêa e Frota [5] para fazer uma correlação entre CIM e a formulação por vértices representantes (VR) e, a partir

das facetas para a formulação VR, derivamos desigualdades válidas para a formulação CIM e estudamos condições em que elas definem facetas. Por fim, iremos estudar melhoras para o processo de geração de colunas - como abreviá-lo e como conseguir limites inferiores no decorrer do processo.

No capítulo 2 definiremos o problema e apresentaremos a notação a ser usada ao longo do texto. Apresentaremos também definições que serão recobradas à medida que os assuntos forem abordados nos capítulos seguintes. Ainda nesse capítulo, falaremos das abordagens para obtenção de limites inferiores para o problema de coloração de vértices e apresentaremos as formulações CIM e VR. No capítulo 3, iremos apresentar as técnicas de geração de linhas e geração de colunas, que irão fundamentar os métodos de resolução das formulações escolhidas. Além disso, uma maneira de combinar a geração de linhas e a geração de colunas é vista nesse capítulo. A seguir, no capítulo 4, relações entre as formulações apresentadas no capítulo 2 são exploradas e correspondências entre as facetas de VR e as desigualdades em CIM serão vistas. Este capítulo também vai basear teoricamente as mudanças propostas neste trabalho para o processo de geração de colunas que resolve a formulação CIM. A descrição dos algoritmos estudados e propostos é o objetivo do capítulo 5. Os diversos algoritmos estão dispostos em pseudo-código para uma rápida referência. No final do capítulo, um algoritmo do macro-processo é mostrado. Adiante vem o capítulo 6, que traz os resultados obtidos em grafos da biblioteca de testes DIMACS [25]. Por fim, o capítulo 7 traz as conclusões do trabalho.

Capítulo 2

O Problema de Coloração

2.1 Definição do Problema

Um grafo G é um par (V, E) onde V é um conjunto finito arbitrário de elementos, chamados vértices, e E é um conjunto de pares não-ordenados de elementos de V , chamados arestas. Os vértices u e v que compõem uma aresta (u, v) são ditos os extremos da aresta. Uma coloração de um grafo $G = (V, E)$ é uma atribuição de cores aos seus vértices, de modo que cada vértice receba pelo menos uma cor e que aos extremos de qualquer aresta de G sejam associados conjuntos disjuntos de cores. Uma coloração usando k cores é dita uma k -coloração. O subconjunto de vértices que recebem uma mesma cor formam uma classe de cor.

O problema de coloração de um grafo G é definido como o problema de encontrar o número mínimo de cores $\chi(G)$, chamado número cromático do grafo G , tal que G admite uma $\chi(G)$ -coloração. Alternativamente, pode ser considerada nesta definição a restrição de que cada vértice receba exatamente uma cor. Observe que as duas definições são equivalentes, uma vez que, para todo $k \geq \chi(G)$, sempre existe uma k -coloração satisfazendo tal condição. Na resolução deste problema, além da determinação do número cromático $\chi(G)$, é desejado que se encontre uma $\chi(G)$ -coloração.

Encontrar uma solução ótima para o problema de coloração é considerado um problema difícil em teoria dos grafos, visto que o mesmo encontra-se na classe NP-difícil [19]. Mais ainda, ele enquadra-se entre os problemas mais desafiadores dessa classe, no sentido de que não permite um esquema de aproximação polinomial, a menos que $P=NP$ [28]. Por outro lado, há diversas aplicações práticas para o problema de coloração, tais como alocação

de frequências [18], escalonamento e quadro de horários escolares [36], aplicações em computação [9] e em matemática [32], dentre outras.

A complexidade de solução do problema e sua vasta gama de aplicações são fatores que têm motivado o interesse pelo seu estudo e o desenvolvimento de vários trabalhos de pesquisa. Apesar dos muitos esforços já dispendidos, os métodos exatos disponíveis para solução do problema ainda não são capazes de resolver instâncias de grande porte. Este trabalho pretende fornecer alguma contribuição nesse sentido.

2.2 Outras Definições e Notações

Para uma melhor compreensão e leitura desse texto, definiremos a seguir a notação a ser usada, bem como apresentaremos conceitos e ferramentas básicos de teoria dos grafos, programação matemática e teoria poliédrica, a serem exploradas nos demais capítulos. Para maiores detalhes sobre os conteúdos das próximas três subseções, sugerimos a leitura das referências [2], [31], [14] e [1].

2.2.1 Teoria dos Grafos

Seja G um grafo, como definido no início da seção 2.1. Adicionalmente, vamos assumir que G não contém laço (aresta com extremos iguais) nem arestas paralelas (entre os mesmos extremos). Os conjuntos de vértices e arestas de G serão denotados por $V(G)$ e $E(G)$, respectivamente, ou apenas por V e E quando o grafo estiver claro pelo contexto. Por simplicidade, vamos denotar uma aresta (u, v) também por uv . O complemento do grafo G é o grafo $\bar{G} = (V, \bar{E})$, onde $\bar{E} = \{(u, v) : (u, v) \notin E, u \neq v\}$. Usaremos n , m e \bar{m} para denotar $|V|$, $|E|$ e $|\bar{E}|$, respectivamente.

Seja $v \in V$ um vértice. Dizemos $u \in V$ é adjacente a ou vizinho de v em G se $(u, v) \in E$. A vizinhança de v , denotada por $N(v)$, é o conjunto de seus vizinhos, ou seja, $N(v) = \{u \in V : (u, v) \in E\}$. A anti-vizinhança de v , denotada por $\bar{N}(v)$, é a vizinhança de v em \bar{G} , isto é, $\bar{N}(v) = \{u \in V : (u, v) \in \bar{E}\}$. A vizinhança e anti-vizinhança fechadas de v são, respectivamente, $N[v] = N(v) \cup v$ e $\bar{N}[v] = \bar{N}(v) \cup v$. Em G , o vértice v é dito universal se $N[v] = V$, isolado se $N(v) = \emptyset$ e dominado se $N(v) \subseteq N(u)$ para algum $u \in V$ diferente de v . Notamos que um vértice isolado é trivialmente um vértice dominado. Para evitar ambigüidades, poderemos usar N_G e \bar{N}_G em vez de N e \bar{N} , respectivamente, quando

quisermos explicitar o grafo de referência.

Um grafo H é dito um subgrafo de $G = (V, E)$ se $V(H) \subseteq V$ e $E(H) \subseteq E$. Dizemos que H é um subgrafo induzido de G se toda aresta de G com extremos em $V(H)$ pertence a $E(H)$, significando que seu conjunto de arestas é $\{(u, v) \in E : u \in V(H), v \in V(H)\}$. Notamos que um subgrafo induzido H de G está bem definido a partir de seu conjunto de vértices, de modo que dizemos ser H induzido por $V(H)$. Em outras palavras, o subgrafo de G induzido por $V' \subseteq V$, que denotamos $G[V']$, tem conjunto de vértices V' e conjunto de arestas $E[V'] = \{(u, v) \in E : u \in V', v \in V'\}$.

Um subgrafo $H \subseteq G$ é maximal com relação a uma certa propriedade se H não é subgrafo próprio de qualquer outro subgrafo $H' \subseteq G$ que satisfaça a propriedade em questão. Pode-se definir subgrafos minimais de maneira análoga.

Um subconjunto de vértices S é um conjunto independente de G se $E[S] = \emptyset$. Por outro lado, $K \subseteq V$ é uma clique de G se $(u, v) \in E[K]$ para todos $u, v \in K$. Por abuso de linguagem, vamos frequentemente confundir uma clique K com o grafo $G[K]$ que ela induz. Vale notar que uma clique em G é um conjunto independente em \bar{G} e vice-versa. A maior cardinalidade de uma clique e de um conjunto independente em G serão denotados por $\omega(G)$ e $\alpha(G)$, respectivamente. O conjunto de todos os conjuntos independentes de G será denotado por $\bar{\mathcal{S}}$, enquanto $\mathcal{S} \subset \bar{\mathcal{S}}$ representará o conjunto dos conjuntos independentes maximais de G .

Vale observar que cliques e conjuntos independentes são estruturas do grafo que geram situações extremas e opostas em relação ao problema de coloração, no seguinte sentido: em qualquer coloração, cada classe de cor é definida por um conjunto independente; em contrapartida, os vértices de uma clique receberão cores distintas entre si, de modo que necessariamente estarão em classes de cor diferentes.

Um caminho em um grafo G é uma seqüência de vértices distintos $P_k = (v_0, \dots, v_k)$ tal que $(v_i, v_{i+1}) \in E(G)$. Um ciclo em G é uma seqüência de vértices $C_k = (v_0, \dots, v_{k-1}, v_k = v_0)$, onde (v_0, \dots, v_{k-1}) é um caminho e (v_{k-1}, v_0) é uma aresta. Comumente, vamos também chamar de caminho P_k e ciclo C_k os subgrafos de G com conjuntos de vértices P_k e C_k e conjuntos de arestas formados por cada aresta entre dois vértices consecutivos. Os tamanhos de P_k e C_k são definidos pelas quantidades de arestas nestes grafos, ou seja, por $|E(P_k)| = |V(P_k)| - 1 = k$ e $|E(C_k)| = |V(C_k)| = k$, respectivamente. Chamamos de corda de um ciclo $C_k = (v_0, \dots, v_{k-1}, v_0)$ uma aresta $(v_i, v_j) \in E$, onde $i > j + 1$.

Se um subgrafo induzido de G por um ciclo não contém cordas, ele é chamado um buraco de G . Um anti-buraco de G é um buraco de \bar{G} . Adicionalmente, buracos e anti-buracos podem ser classificados quanto à paridade de seu conjunto de vértices. Um (anti-)buraco é par se possui um número par de vértices ou ímpar se possui um número ímpar de vértices.

Um grafo é conexo se para todo par de vértices u e v existe um caminho (v_1, \dots, v_k) tal que $v_1 = u$ e $v_k = v$. Uma componente de um grafo é um subgrafo maximal conexo.

Uma orientação de G é a definição de uma função $\sigma : E \rightarrow V$ tal que $\sigma(uv) \in \{u, v\}$. Se $\sigma(uv) = v$, diz-se que a aresta está orientada de u para v . Uma orientação é acíclica se em qualquer ciclo de G as arestas não estão todas orientadas em um mesmo sentido. Notamos que, em particular, uma ordenação \prec dos vértices de G define uma orientação acíclica de G tal que $\sigma(uv) = v$ se, e somente se, $u \prec v$. Tal ordenação define igualmente uma orientação de \bar{G} .

Considerando que nosso foco é o problema de coloração, vamos descartar certas situações triviais, assumindo hipóteses sobre G . Primeiro, observamos que se G_1, G_2, \dots, G_r são as componentes conexas de G , então $\chi(G) = \max\{\chi(G_i) : i = 1, 2, \dots, k\}$. Segundo, vértices universais e dominados são irrelevantes com respeito ao problema de coloração no seguinte sentido. Se v é universal, então $\chi(G) = \chi(G \setminus \{v\}) + 1$, posto que v necessariamente forma uma classe de cor em toda coloração possível. Já se v é dominado, ocorre que $\chi(G) = \chi(G \setminus \{v\})$, pois v pode sempre receber uma color de um vértice que o domina. Nesse sentido, ao longo do desenvolvimento desse texto, vamos considerar as seguintes hipóteses, que podem ser facilmente verificadas.

H1 G é conexo.

H2 G não possui vértices universais ou dominados.

2.2.2 Teoria Poliédrica

Um poliedro é um conjunto $P \subseteq \mathbb{R}^q$, $q \in \mathbb{Z}_+$ definido pela interseção de um número finito p de semi-espacos em \mathbb{R}^q , onde, por sua vez, um subespaço é um conjunto $\{x \in \mathbb{R}^q : \alpha^x \leq \alpha_0\}$ com $\alpha \in \mathbb{R}^{1 \times q} \setminus \{0\}$ e $\alpha_0 \in \mathbb{R}$. Sendo assim, um poliedro pode ser descrito como:

$$\{x \in \mathbb{R}^q : Ax \leq b\},$$

onde $A \in \mathbb{R}^{p \times q}$ e $b \in \mathbb{R}^p$, $p \in \mathbb{Z}_+$. É fácil perceber que existem várias possibilidades para A e b na descrição acima. Para caracterizarmos um sistema minimal de desigualdades que definem P , necessitamos introduzir outros conceitos.

Um vetor $x \in \mathbb{R}^q$ é uma combinação linear dos vetores $x^1, \dots, x^k \in \mathbb{R}^q$ se

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

para algum $\alpha \in \mathbb{R}^k$. Se $\sum_{i=1}^k \alpha_i = 1$, a combinação linear é dita afim, e se, adicionalmente, $\alpha \geq 0$ é denominada convexa. O fecho convexo de um conjunto não-vazio $X \subseteq \mathbb{R}^q$, denotado por $\text{conv}(X)$, é o conjunto de todos os vetores que são combinação convexa de um subconjunto finito de X . Um politopo é um poliedro limitado. Podemos dizer também que um politopo é o fecho convexo de um número finito de pontos.

Um conjunto $X \subseteq \mathbb{R}^q$ é dito linearmente independente se, para qualquer subconjunto finito $X' \subseteq X$, 0 é combinação linear de X' se, e somente se, $\alpha = 0$. Dizemos ainda que X é afim-independente se, para qualquer subconjunto finito $\{x^1, \dots, x^k\} \subseteq X$, a igualdade $\sum_{i=1}^k \alpha_i x^i = 0$, com $\sum_{i=1}^k \alpha_i = 0$, implica que $\alpha = 0$.

Para $X \subseteq \mathbb{R}^q$, a dimensão de X , denotada por $\text{dim}(X)$, é a cardinalidade de um maior subconjunto de X que é linearmente independente. Mostra-se que $\text{dim}(X)$ é uma unidade a menos que a cardinalidade de um maior conjunto afim-independente contido em X . Dizemos que $X \subseteq \mathbb{R}^q$ tem dimensão plena se $\text{dim}(X) = q$. O posto de uma matriz A , denotado por $\text{posto}(A)$, é a dimensão do conjunto de vetores-coluna de A , que é igual à dimensão do conjunto de vetores-linha de A .

Dados $\alpha \in \mathbb{R}^{1 \times q} \setminus \{0\}$ e $\alpha_0 \in \mathbb{R}$, a desigualdade $\alpha x \geq \alpha_0$ é válida para o poliedro $P \subseteq \mathbb{R}^q$ se $P \subseteq \{x \in \mathbb{R}^q : \alpha x \geq \alpha_0\}$. Sejam $\alpha x \geq \alpha_0$ e $\beta x \geq \beta_0$ desigualdades válidas para $P \subseteq \mathbb{R}^q$. Se $(\alpha, \alpha_0) = k(\beta, \beta_0)$, para algum $k \in \mathbb{R}$, $k > 0$, as desigualdades são ditas equivalentes. Do contrário, se $\{x \in \mathbb{R}^q : \alpha x \geq \alpha_0\} \subseteq \{x \in \mathbb{R}^q : \beta x \geq \beta_0\}$, dizemos que a primeira desigualdade domina a segunda. Em particular, quando $P \subseteq \mathbb{R}_+^q$, então $\alpha x \geq \alpha_0$ domina $\beta x \geq \beta_0$ se existe $k > 0$ tal que $\alpha \leq k\beta$, $\alpha_0 \geq k\beta_0$ e $(\alpha, \alpha_0) \neq k(\beta, \beta_0)$.

Um conjunto $F \subseteq P$ é uma face de P se existe uma desigualdade $\alpha x \geq \alpha_0$, válida para P , tal que $F = P \cap \{x \in \mathbb{R}^q \mid \alpha^T x = \alpha_0\}$. Dizemos que uma face F é uma faceta de P se F não está propriamente contida em nenhuma outra face não-vazia $F' \neq P$. Neste caso, dizemos que a desigualdade $\alpha^T x \geq \alpha_0$ define a faceta F . Mostra-se que um sistema minimal para descrever P é dado por desigualdades válidas que definem facetas de P .

Caracterizações para facetas de poliedros de dimensão plena são dadas pelo teorema a seguir [14].

Teorema 1. *Seja $P \subseteq \mathbb{R}^q$ um poliedro de dimensão plena e $F = \{x \in P \mid \alpha^T x = \alpha_0\}$ uma face de P , com $F \neq P$. Então, as seguintes assertivas são equivalentes:*

1. F é uma faceta de P ;
2. $\dim(F) = q - 1$;
3. se $\beta^T x \geq \beta_0$, $\beta \neq 0$, é uma desigualdade válida e $F \subseteq \{x \in P \mid \beta^T x = \beta_0\}$, então existe $k \in \mathbb{R}$, $k > 0$, tal que $\beta = k\alpha$ e $\beta_0 = k\alpha_0$.

Do teorema acima conclui-se que uma desigualdade válida define uma faceta de P se, e somente se, ela não pode ser escrita como combinação linear positiva de outras desigualdades válidas para P . Além disso, temos que uma desigualdade válida dominada não define facea. Estas serão as estratégias que usaremos para mostrar se uma certa desigualdade válida define ou não uma faceta.

2.2.3 Otimização Matemática

Um problema de otimização matemática consiste em minimizar (ou maximizar) uma função, chamada função objetivo, dentro de um subconjunto do seu domínio, denominado conjunto viável. Em nosso contexto, vamos considerar apenas funções objetivo assumindo valores reais (finitos). Assim, podemos escrevê-lo da seguinte forma:

$$(PO) \min f(x) \tag{2.1a}$$

$$\text{s.a: } x \in X, \tag{2.1b}$$

onde $f : Y \rightarrow \mathbb{R}$ e $X \subseteq Y$. Em geral, o conjunto X é definido por um grupo de equações ou inequações, que são chamadas restrições de (PO). Um ponto $x \in X$ é dito solução viável, enquanto uma solução ótima é uma solução viável x^* tal que $f(x^*) \leq f(x)$, para todo $x \in X$. O valor de uma solução $x \in X$ é $f(x)$. Se x^* é a solução ótima, então $f(x^*)$ é o valor ótimo de (PO).

Associado ao problema (PO), seja dado um outro problema de otimização (PO'), definido por uma função objetivo $f' : Y' \rightarrow \mathbb{R}$ e um conjunto viável $X' \subseteq Y'$. Dizemos que (PO') é uma relaxação de (PO) ou, equivalentemente, (PO) é um problema restrito

com relação a (PO') se $X \subseteq X'$ e $f'(x) \leq f(x)$, para todo $x \in X$. Particularmente, descartar algumas restrições de (PO) define uma relaxação, enquanto fixar valores para algumas variáveis de (PO) determinam um problema restrito. Observamos que o valor ótimo de uma relaxação fornece um limite inferior para o valor ótimo do problema original, enquanto um limite superior para ele é dado pelo valor de qualquer solução viável de um problema restrito.

Um problema de programação linear é um problema de otimização onde a função objetivo é linear e o conjunto viável é um poliedro. Pode-se mostrar que qualquer problema de programação linear pode ser escrito como a seguir:

$$(PPL) \min cx \tag{2.2a}$$

$$\text{s.a: } Ax = b \tag{2.2b}$$

$$x \geq 0 \tag{2.2c}$$

onde $x \in \mathbb{R}^q$, $A \in \mathbb{R}^{p \times q}$, $c \in \mathbb{R}^{1 \times q}$ e $b \in \mathbb{R}^p$, com $\text{posto}(A) = p$. À (2.2b) chamamos restrições do PPL. Já (2.2c) definem as restrições de não-negatividade.

A cada problema de programação linear associa-se um outro problema de programação linear, chamado de problema dual e definido como:

$$(DUAL) \max ub \tag{2.3a}$$

$$\text{s.a: } uA \leq c \tag{2.3b}$$

A variável $u \in \mathbb{R}^p$ é chamada de variável dual de PPL. Observe que para cada restrição em PPL, associamos uma variável em DUAL, e a cada variável em PPL temos uma restrição em DUAL.

O problema dual possui muitas propriedades com respeito ao problema original, chamado de primal [1]. As relações entre os dois problemas começam pelos valores das soluções viáveis de cada um deles.

Teorema 2 (Teorema Fraco da Dualidade). *Se x satisfaz as restrições de PPL e u satisfaz as restrições de DUAL então $cx \geq ub$.*

Uma consequência da relação acima é que as soluções são ótimas se uma igualdade ocorre. Essa condição também é necessária para a otimalidade.

Teorema 3 (Teorema Forte da Dualidade). *Sejam x^* e u^* satisfazendo as restrições de PPL e DUAL, respectivamente. Então, x^* e u^* são soluções ótimas se, e somente se, $cx^* = u^*b$.*

Soluções ótimas de PPL e DUAL podem ser caracterizadas a partir de bases da matriz A , ou seja, submatrizes de A , de ordem p , que são invertíveis. Nesse sentido, seja dada uma partição da matriz A em $A = [B \ N]$, onde B é uma base. Vamos denotar por x_B e x_N as variáveis associados a B e N no sistema $Ax = b$ e chamá-las de variáveis básicas e não-básicas, respectivamente. De forma análoga, vamos considerar a partição de $c = [c_B \ c_N]$. Dizemos que x é uma solução básica (relativa a B) de PPL se $x_N = 0$ e, conseqüentemente, $x_B = B^{-1}b$. Dizemos ainda que $u = c_B B^{-1}$ é uma solução básica (relativa a B) para DUAL. O vetor $\bar{c} = c - uA$ é denominado vetor de custos reduzidos. Observamos x e u são viáveis se, e somente se, $x_B \geq 0$ e $\bar{c} \geq 0$. As seguintes propriedades podem ser encontradas em [1].

Teorema 4. *Se PPL tem solução ótima, pelo menos um delas é solução básica.*

Teorema 5. *Sejam B uma base de A e x e u as soluções básicas definidas por B . Se $x_B \geq 0$ e $\bar{c} \geq 0$ então x e u são soluções ótimas.*

Um segundo tipo de problema de otimização que será abordado nesse trabalho é o problema de programação inteira, que é definido a partir de PPL, restringindo algumas variáveis a assumirem apenas valores inteiros. Essas restrições adicionais são chamadas de restrições de integralidade. Formalmente, um problema de programação inteira é definido como:

$$\text{(PPI) } \min \quad c_1x_1 + c_2x_2 \quad (2.4a)$$

$$\text{s.a: } \quad A_1x_1 + A_2x_2 = b \quad (2.4b)$$

$$x_1 \geq 0, \quad x_2 \geq 0 \quad (2.4c)$$

$$x_2 \in Z^{q_2}. \quad (2.4d)$$

Enquanto um problema de programação linear é classificado como polinomial, um problema de programação inteira é NP-difícil. Para a sua solução, por vezes são utilizadas técnicas que consideram relaxações ou problemas restritos de PPI, como forma de determinar limites inferiores e superiores. Uma relaxação usual, chamada relaxação

linear, é o PPL obtido quando desconsideramos as restrições de integralidade. Por outro lado, a fixação de valores de algumas variáveis é estratégia comum para a geração de subproblemas restritos. Para maiores detalhes sobre o uso de relaxações e problemas restritos na solução de um problema de programação inteira, sugerimos uma consulta a [34].

Outra abordagem usual para a resolução de PPI baseia-se no seguinte resultado [31].

Proposição 1. *Seja X o conjunto viável de PPI. Então pode-se afirmar que:*

1. *$\text{conv}(X)$ é um poliedro,*
2. *PPI é equivalente ao problema de programação linear $\min\{cx : x \in \text{conv}(X)\}$.*

À luz da proposição acima, uma estratégia para resolver PPI consiste em explorar o programa linear definido por $\text{conv}(X)$. Em referências a esse contexto, vamos chamar de poliedro (ou politopo, se limitado) associado a uma formulação de programação inteira o fecho convexo de seu conjunto viável.

Uma dificuldade com essa abordagem está no fato de que encontrar o sistema de desigualdades que descreve $\text{conv}(X)$ é usualmente tão difícil quanto resolver PPI. Sendo assim, na prática, a estratégia consiste em aproximar $\text{conv}(X)$ o melhor possível, especialmente na região em torno da solução ótima, o que é feito de preferência pela determinação de facetas.

É fácil constatar que o problema de coloração é um exemplo de problema de otimização, onde X descreve conjunto de todas as possíveis colorações de G e $f(x)$ é o número de cores na coloração $x \in X$. Na verdade, como veremos da seção 2.3, este problema pode ser descrito através de várias formulações tipo PPI.

2.3 Formulações de Programação Inteira

Trabalhos recentes de diversos autores incentivam o estudo do problema de coloração a partir de formulações de programação inteira [4, 5, 11, 13, 16, 29]. A seguir faremos breves comentários sobre as formulações propostas na literatura e, em seguida, detalharemos duas delas, que serão exploradas em nosso trabalho. Comparações entre essas formulações podem ser encontradas em [8].

A formulação clássica para o problema de coloração usa uma variável binária associada a cada possível cor e a cada vértice do grafo. Para definir uma coloração, se um

determinado vértice recebe uma certa cor, a variável corresponde assume valor 1; caso contrário lhe é atribuído valor 0. Essa formulação utiliza um número polinomial de variáveis e restrições. Diaz e Zabala [13] e Coll *et al.* [11] fortalecem essa formulação com o acréscimo de restrições adicionais e definição de facetas e a utilizam para o desenvolvimento de um algoritmo do tipo *branch-and-cut*, que conta adicionalmente com uma etapa de pré-processamento para reduzir o tamanho do grafo.

Já em [16], Figueredo *et al.* propõem uma formulação baseada na correspondência entre colorações e orientações acíclicas de um grafo. Uma simplificação dessa formulação é, em seguida, apresentada por Figueredo [15]. Tanto a versão original quanto a simplificada são descritas por um número exponencial de restrições, o que deve ser cuidadosamente tratado em uma implementação computacional. Também em [15], um estudo parcial dos politopos associados a ambas as formulações é apresentado.

Mehrotra e Trick utilizam, em [29], uma formulação que associa uma variável binária a cada conjunto independente maximal do grafo. A variável recebe valor 1 se o conjunto independente associado é escolhido para definir uma classe de cor. Como o número de conjuntos independentes do grafo é exponencial, um método de geração de colunas é proposto pelos autores, que o incorporam em um algoritmo *branch-and-bound*. Esse trabalho alcançou os melhores resultados em termos de limite inferior para o problema de coloração.

Uma quarta formulação para o problema foi recentemente proposta em [5] por Campêlo, Corrêa e Frota, usando a idéia de vértice representante de uma classe de cor. Essa formulação possui um número polinomial tanto de restrições quanto de variáveis. Além de propor a formulação, neste mesmo trabalho, os autores desenvolvem o estudo parcial do politopo associado.

Essas duas últimas formulações serão estudadas neste trabalho, motivo pelo qual as descreveremos em detalhes nas próximas subseções. O objetivo do estudo é investigar relações entre elas, que viabilizem, entre outras idéias, a combinação da formulação por conjuntos independentes maximais (CIM) com as estruturas faciais descritas para a formulação por vértices representantes de cor (VR). Com isso, pretende-se fortalecer ainda mais a formulação CIM.

2.3.1 Formulação por Conjuntos Independentes Maximais (CIM)

Uma coloração de G pode ser vista como uma divisão de V em uma família $C = \{S_1, \dots, S_k\}$ de $k \geq \chi(G)$ conjuntos independentes de G , onde cada conjunto independente está associado a uma cor. Na verdade, podemos assumir que os conjuntos da família são maximais quando permitimos multicoloração dos vértices. Em outras palavras, uma coloração é uma atribuição $\phi : \mathcal{S} \rightarrow \{0, 1\}$ tal que

$$\sum_{S \in \mathcal{S}: v \in S} \phi(S) \geq 1, \quad \forall v \in V. \quad (2.5)$$

Notamos que, dessa forma, $C = \{S \in \mathcal{S} : \phi(S) = 1\}$.

Nessa perspectiva, uma formulação para o problema de coloração é obtida associando uma variável binária $\lambda_S \in \mathcal{B}$ a cada conjunto independente maximal $S \in \mathcal{S}$. Considerando que $\lambda_S = 1$ se, e somente se, o conjunto independente S é escolhido como uma classe de cor, a formulação é então:

$$\text{(CIM) } \min \sum_{S \in \mathcal{S}} \lambda_S \quad (2.6a)$$

$$\text{s.a: } \sum_{S \in \mathcal{S}: v \in S} \lambda_S \geq 1, \quad \forall v \in V \quad (2.6b)$$

$$\lambda_S \in \{0, 1\}, \quad \forall S \in \mathcal{S} \quad (2.6c)$$

A restrição (2.6b) estabelece que todo vértice deve ser coberto por pelo menos um conjunto independente, ou seja, pertence a uma classe de cor. A restrição de que dois vértices adjacentes recebem cores distintas é satisfeita pela própria definição das variáveis. Nesta formulação temos um número exponencial de variáveis e $O(n)$ restrições.

Notamos que, se cada vértice é restrito a receber somente uma cor, a família C é formada por conjuntos disjuntos, definindo, portanto, uma partição de V . Além disso, os conjuntos independentes possivelmente não serão maximais. Para este caso, devemos substituir o conjunto \mathcal{S} por $\bar{\mathcal{S}}$ na formulação acima, bem como a desigualdade em (2.6b) por uma igualdade.

O fato de a formulação (2.6) ter um número exponencial de variáveis pode parecer a princípio desencorajador. Entretanto, ela pode ser habilmente tratada, como fazem Mehrotra e Trick [29], utilizando uma técnica de geração de colunas, a ser vista na seção 3.1.

Vale ainda destacar que a relaxação linear de (2.6) fornece o número cromático fracionário $\chi_F(G)$ do grafo G , que é definido como a seguir. Uma coloração fracionária de G é uma atribuição $\phi : \mathcal{S} \rightarrow [0, 1]$ satisfazendo (2.5). O número cromático fracionário de G é

$$\chi_F(G) = \min \left\{ \sum_{S \in \mathcal{S}} \phi(S) : \phi \text{ é uma coloração fracionária} \right\} \quad (2.7)$$

Portanto, $\chi_F(G)$ é o valor ótimo de 2.6a quando relaxamos as condições de integralidade 2.6c.

2.3.2 Formulação por Vértices Representantes de Cor (VR)

Novamente, vamos considerar uma coloração como uma família de conjuntos independentes $C = \{S_1, S_2, \dots, S_k\}$. Suponha que escolhemos um vértice em S_i para representar a cor $i = 1, \dots, k$. Esse vértice será chamado representante. Logo, em C , cada vértice v se representa ou é representado por algum vértice em $\bar{N}(v)$. A partir desta observação, Campêlo, Corrêa e Frota [5] desenvolveram uma formulação de programação inteira para o problema de coloração, usando variáveis binárias x_{uv} , para todo $u \in V$ e $v \in \bar{N}[u]$, com a seguinte interpretação: $x_{uv} = 1$ se, e somente se, u representa a cor de v . Assim, um vetor x^* contendo todas essas variáveis é o vetor de incidência de uma coloração de valor $\chi(G)$ se e, somente se, ele é solução de:

$$(VR) \min \sum_{v \in V} x_{vv} \quad (2.8a)$$

$$\text{s.a:} \quad \sum_{u \in \bar{N}[v]} x_{uv} \geq 1, \quad \forall v \in V \quad (2.8b)$$

$$x_{uv} + x_{uw} \leq x_{uu}, \quad \forall u \in V \text{ e } \forall vw \in E[\bar{N}(u)] \quad (2.8c)$$

$$x_{uv} \in \{0, 1\}, \quad \forall v \in V, \forall u \in \bar{N}[v] \quad (2.8d)$$

Para utilizarmos o menor número de cores, minimizamos em 2.8a a quantidade de representantes. As restrições (2.8b) indicam que todo vértice deve ser representado por pelo menos um vértice, ou seja, deve pertencer a alguma classe de cor. Já as restrições (2.8c) significam que um vértice não pode representar dois vértices adjacentes. Observe que as restrições $x_{uv} \leq x_{uu}$, para todo $u \in V$ e $v \in \bar{N}(u)$, que asseguram que só um vértice representante pode representar outro vértice, decorrem de (2.8c) pela hipótese H2. Tal hipótese garante a não existência de vértice isolado v em $\bar{N}(u)$. Nesta formulação, temos $O(n + \bar{m})$ variáveis e $O(n\bar{m})$ restrições.

Também em [5], os autores realizam um estudo parcial do politopo associado a (VR), derivando duas classes de facetas baseadas em subgrafos de G , que incluem cliques, buracos, anti-buracos.

Podemos observar que a formulação (2.8) possui simetria, no sentido de que uma mesma classe de cor S_i pode ser representada por $|S_i|$ diferentes maneiras. Para resolver este problema, Campelo, Corrêa e Frota [6] consideram uma ordenação total \prec dos vértices de V . Para cada $v \in V$, definem $\bar{N}^+(v) = \{u \in \bar{N}(v) | v \prec u\}$, $\bar{N}^+[v] = \bar{N}^+(v) \cup \{v\}$, $\bar{N}^-(v) = \bar{N}(v) \setminus \bar{N}^+(v)$, e $\bar{N}^-[v] = \bar{N}^-(v) \cup \{v\}$. Se u e v são não-adjacentes em G , u está autorizado a representar v se, e somente se, $v \in \bar{N}^+(u)$. Com essas restrições, a formulação por vértices representantes de cor torna-se:

$$(VRA) \min \sum_{v \in V} x_{vv} \quad (2.9a)$$

$$\text{s.a:} \quad \sum_{u \in \bar{N}^-[v]} x_{uv} \geq 1, \quad \forall v \in V \quad (2.9b)$$

$$x_{uv} + x_{uw} \leq x_{uu}, \quad \forall u \in V \text{ e } vw \in E[\bar{N}^+(u)] \quad (2.9c)$$

$$x_{uv} \leq x_{uu}, \quad \forall u \in V \text{ e } v \text{ isolado em } \bar{N}^+(u) \quad (2.9d)$$

$$x_{uv} \in \{0, 1\}, \quad \forall v \in V, \forall u \in \bar{N}^-[v] \quad (2.9e)$$

Um estudo do politopo associado a esta nova versão da formulação por representantes de cor é apresentada por Campêlo, Campos e Corrêa [4]. Na verdade, os autores adaptam e generalizam as duas classes de facetas derivadas em [5], bem como apresentam uma terceira classe de facetas baseada em conjuntos independentes do grafo.

2.4 Abordagens para o Limite Inferior

Métodos exatos para determinar o número cromático de um grafo, um problema, como vimos, NP-difícil, demandam, via de regra, o cálculo de bons limites inferiores e superiores. Por isso, o estudo desses elementos faz-se importante. Neste trabalho iremos explorar o estudo do limite inferior para o problema de coloração.

Dentre as abordagens usais para determinar um limite inferior para $\chi(G)$, duas merecem destaque. A primeira baseia-se em métodos de decomposição estrutural do grafo [21], nos quais se procura decompor o grafo em subgrafos menores e possivelmente mais fáceis de se colorir, bem como identificar subestruturas críticas para a determinação de $\chi(G)$.

Uma outra possível abordagem é o uso da programação matemática, onde se explora uma formulação de programação inteira para o problema. Essa abordagem vem ganhando força mais recentemente, como mostram vários trabalhos publicados nos últimos dez anos [4, 5, 11, 13, 16, 29].

2.4.1 Subgrafos Críticos

Um método para a obtenção de um limite inferior para o número cromático de G consiste em pesquisar subgrafos de G para os quais saibamos determinar uma coloração ótima, conforme justifica a seguinte proposição.

Proposição 2. *Para todo $H \subseteq G$, $\chi(G) \geq \chi(H)$.*

Prova: Se um grafo G é colorido com $\chi(G)$ cores, um subgrafo $H \subseteq G$ pode ser colorido repetindo-se as cores utilizadas em G , utilizando assim no máximo $\chi(G)$ cores. \square

Os subgrafos minimais de G que fornecem um limite inferior k para $\chi(G)$ são definidos por grafos k -críticos. Um grafo H é dito k -crítico se $\chi(H) = k$ e qualquer subgrafo de H tem número cromático menor que $\chi(H)$. Em geral, chamamos de subgrafo crítico de G a um subgrafo k -crítico H , para qualquer $k > 0$. Como exemplos de grafos críticos, podemos citar os buracos ímpares (grafos 3-críticos), os anti-buracos ímpares de tamanho k (grafos $(\frac{k-1}{2})$ -críticos) e as cliques de tamanho k (grafos k -críticos).

A partir da Proposição 2, podemos concluir que o tamanho de qualquer clique de G fornece um limite inferior para $\chi(G)$, posto que dois vértices de uma clique não podem receber a mesma cor. Dessa observação, temos o seguinte corolário:

Corolário 1. $\chi(G) \geq \omega(G)$.

O limite fornecido pelo corolário acima pode não ser exato. Por exemplo, quando G é um buraco ímpar, temos que $\omega(G) = 2$ e $\chi(G) = 3$. Na verdade, o Teorema Forte dos Grafos Perfeitos estabelece que, se $\chi(G) > \omega(G)$, então G contém um buraco ou anti-buraco ímpar. Mais ainda, o número cromático pode ser arbitrariamente maior que o tamanho da maior clique. Essa situação pode ser demonstrada construindo grafos através de operações de Mycielski.

Dado um grafo $F = (V, E)$, com $V = \{v_1, \dots, v_n\}$, uma operação de Mycielski sobre

F contrói um grafo $M(F) = (V', E')$ tal que

$$V' = V \cup U \cup \{z\}, \text{ sendo } U = \{u_1, \dots, u_n\}, \quad (2.10)$$

$$E' = E \cup \{(u_i, v_j) : (v_i, v_j) \in E\} \cup \{(z, u_i) : u_i \in U\}. \quad (2.11)$$

Em particular, a aplicação iterativa dessa operação, a partir de uma aresta, define a classe de grafos de Mycielski [30]. Precisamente, o k -ésimo grafo de Mycielski M_k é definido recursivamente como $M_k = M(M_{k-1})$, para $k \geq 3$, sendo M_2 o grafo formado por dois vértices ligados por uma aresta e M_1 o grafo definido por um único vértice. Na figura 2.1, ilustramos o grafo M_4 , que é igual a $M(C_5)$.

Figura 2.1: Grafo $M_4 = M(C_5)$

Grafos contruídos através de operações de Mycielski possuem as seguintes propriedades.

Proposição 3. *Dados um grafo F , com pelo menos uma aresta, e o grafo $M(F)$ gerado pela aplicação da operação de Mycielsky em F , temos:*

P1 $\chi(M(F)) = \chi(F) + 1$

P2 $\omega(M(F)) = \omega(F)$

P3 *Se F é k -crítico, então $M(F)$ é $(k + 1)$ -crítico.*

Prova: As propriedades P1 e P2 estão demonstradas em [26]. Considerando a notação usada em (2.10)–(2.11), vamos provar a propriedade P3, mostrando que $\chi(M(F) \setminus \{v\}) =$

$\chi(F)$, para todo $v \in V \cup U \cup \{z\}$. Primeiro, observamos que, sendo F k -crítico, para todo $v_i \in V$, sempre existe uma $\chi(F)$ -coloração onde uma classe de cor é exatamente $\{v_i\}$. Denotamos essa coloração por C_i . Segundo, em $M(F)$,

$$N(v_i) = N(u_i) \cup \{z\}, \quad \forall i = 1, 2, \dots, n, \quad (2.12)$$

$$N(z) = U \quad (2.13)$$

e U é um conjunto independente. Seja $i \in \{1, 2, \dots, n\}$. Por (2.12), uma coloração ótima C_i de F pode ser estendida para $M(F) \setminus \{z\}$, colorindo $u_i \in U$ com a mesma cor de $v_i \in V$. Isto mostra que $\chi(M(F) \setminus \{z\}) = \chi(F)$. Adicionalmente, $M(F) \setminus \{u_i\}$ pode ser colorido a partir de C_i , atribuindo a z a cor de v_i e a $u_j, j \neq i$, a cor de v_j , o que implica $\chi(M(F) \setminus \{u_i\}) = \chi(F)$. Finalmente, para obtermos $\chi(M(F) \setminus \{v_i\}) = \chi(F)$, usamos C_i e colorimos todos os vértices de U com a cor ora atribuída a v_i e incluímos z na mesma classe de algum $v_j, j \neq i$. \square

Em nosso estudo vamos nos concentrar em subgrafos críticos do tipo buracos e anti-buracos ímpares, cliques maximais e grafos gerados a partir de operações recursivas de Mycielsky sobre esses subgrafos.

2.4.2 Programação Matemática

A abordagem para obter limites inferiores para o número cromático do grafo G , usando programação inteira, fundamenta-se essencialmente no seguinte resultado:

Proposição 4. *Se Π é uma formulação de programação inteira para o problema de coloração então o valor ótimo de uma relaxação de Π é um limite inferior para $\chi(G)$.*

Assim, a partir de (2.7) e somando as restrições (2.6b) para os vértices de uma clique máxima, mostramos que:

Corolário 2. $\chi(G) \geq \chi_F(G) \geq \omega(G)$.

Embora, pelo corolário 2, o número cromático fracionário seja um limite inferior melhor que o tamanho de uma cliquemáxima, ainda pode acontecer de $\chi(G)$ ser arbitrariamente maior que $\chi_F(G)$. Novamente, isto acontece para grafos contruídos por operações de Mycielski, como podemos concluir do seguinte resultado demonstrado por Larsen, Propp e Ullman [26].

Proposição 5. *Se H é um grafo com pelo menos uma aresta e $M(H)$ o grafo gerado pela aplicação da operação de Mycielsky em H então $\chi_F(M(H)) = \chi_F(H) + \frac{1}{\chi_F(H)}$.*

Por outro lado, considerando todas as formulações apresentadas na seção 2.3 e suas relaxações lineares, pode-se mostrar que o melhor limite inferior é fornecido pela formulação (CIM), ou seja, $\chi_F(G)$. Para se ultrapassar esse limite é preciso a adição de desigualdades válidas que possam tornar a relaxação mais apertada.

Embora as classes de facetas conhecidas para as outras três formulações tenham sido usadas para fortalecer as formulações correspondentes, os resultados computacionais conhecidos não são capazes, em geral, de superar o limite dado por $\chi_F(G)$. Sendo assim, vamos investir neste trabalho em derivar desigualdades válidas para (CIM) que possam fornecer um melhor limite inferior. Para isso, usaremos resultados conhecidos para a formulação (VR).

Capítulo 3

Métodos de Solução

Neste capítulo vamos abordar métodos de solução de problemas de programação linear com um número muito grande de variáveis ou restrições ou ainda quando os coeficientes que definem o problema não são conhecidos a priori. Essas situações ocorrem naturalmente em alguns problemas ou podem aparecer a partir de manipulações da formulação original. Esse segundo caso ocorre, por exemplo, quando realizamos uma decomposição de Dantzig-Wolfe ou quando procuramos aproximar o poliedro associado a uma formulação de programação inteira.

3.1 Geração de Colunas

3.1.1 Princípio da Geração de Colunas

Seja (P) o seguinte problema de programação linear com muitas variáveis:

$$(P) \min cx \tag{3.1a}$$

$$\text{s.a: } Ax \leq b \tag{3.1b}$$

$$x \geq 0 \tag{3.1c}$$

onde possivelmente nem se conhece a matriz A ou o vetor c explicitamente. Por exemplo, em algumas aplicações, as colunas de A advêm da solução de um outro problema.

Para resolver (P) , podemos começar com um subconjunto de variáveis (e de colunas de $\begin{bmatrix} c \\ A \end{bmatrix}$) e resolver o problema restrito. Obtemos assim uma solução viável para (P) . Então, usamos as variáveis duais π , associadas a (3.1b), para resolver o subproblema:

$$\delta = \min_j c_j - \mu A_j \quad (3.2a)$$

$$\text{s.a: } \begin{bmatrix} c_j \\ A_j \end{bmatrix} \in \text{col}(c, A) \quad (3.2b)$$

onde $\text{col}(c, A)$ representa o conjunto das colunas de $\begin{bmatrix} c \\ A \end{bmatrix}$.

Caso $\delta \geq 0$, a solução do problema restrito fornece a solução de (P). Do contrário a coluna $\begin{bmatrix} c_{j^*} \\ A_{j^*} \end{bmatrix}$, solução do subproblema 3.2, é acrescentada ao problema restrito. Para que essa estratégia possa ser empregada, é necessário que a solução do subproblema 3.2 seja factível.

Enquadra-se na situação exposta acima, por exemplo, a relaxação linear da formulação do problema de coloração por conjuntos independentes maximais (2.6). De fato, a matriz de restrições possui uma quantidade de colunas que pode ser uma função exponencial do número de vértices do grafo G , e cada coluna é o vetor característico de um conjunto independente de G . Neste caso, o problema para gerar a coluna é $\min_z 1 - \pi z$, onde $z \in \{0, 1\}^n$ descreve um conjunto independente de G . Tal problema, denominado Problema do Conjunto Independente Máximo Ponderado, pode ser formulado como:

$$\text{(CIMP) } \max \sum_{v \in V} \pi_v z_v \quad (3.3a)$$

$$\text{s.a: } z_v + z_u \leq 1, \forall (u, v) \in E \quad (3.3b)$$

$$z_v \in \{0, 1\}, \forall v \in V \quad (3.3c)$$

Se o valor ótimo desse problema for maior que 1, então $S = \{v \in V | z_v = 1\}$ é o conjunto independente cuja coluna deve ser adicionada. Caso contrário, a solução da relaxação linear de (2.6) foi encontrada.

Começando com uma família inicial \mathcal{S}' de conjuntos independentes, é executado o procedimento acima até que não haja mais conjuntos independentes que melhorem a solução. Se o resultado da relaxação retorna um λ_S inteiro para todo $S \in \mathcal{S}'$, então temos a solução ótima para o problema (CIM). Caso contrário, teremos que forçar a integralidade das variáveis. Este processo é implementado em [29].

Outra situação importante para a aplicação do método de geração de colunas ocorre quando da decomposição de um problema de programação linear ou inteira segundo o princípio de Dantzig-Wolfe, conforme seção a seguir.

3.1.2 Decomposição de Dantzig-Wolfe

O princípio da decomposição de Dantzig-Wolfe visa a particionar um problema de programação linear ou linear-inteira, de modo a facilitar sua resolução, explorando a estrutura das restrições. Com a decomposição, gera-se um problema mestre e um ou mais subproblemas escravos. O problema mestre, que é equivalente ao problema decomposto, apresenta um número menor de restrições, mas um número grande de variáveis, que serão tratadas implicitamente. Colunas do novo problema serão geradas através da solução dos problemas escravos, que, geralmente, são construídos a partir das restrições originais que podem ser tratadas com certa facilidade. Esse princípio de decomposição, proposto por Dantzig e Wolfe [35], é detalhado a seguir.

Seja (P) o seguinte problema de programação inteira:

$$(P) \min cx \tag{3.4a}$$

$$\text{s.a: } Ax = b \tag{3.4b}$$

$$Dx \leq d \tag{3.4c}$$

$$x \in Z_+^n \tag{3.4d}$$

onde $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m_1 \times n}$, $D \in \mathbb{R}^{m_2 \times n}$, $b \in \mathbb{R}^{m_1}$, $d \in \mathbb{R}^{m_2}$

Por simplicidade, assumiremos que $H = \{x \in Z_+^n | Dx \leq d\}$ é um conjunto finito de elementos $\{x^1, \dots, x^p\}$, onde $p = |H|$. Existe uma correspondência um-a-um entre os elementos de H e as soluções de

$$x = \sum_{k=1}^p \lambda_k x^k; \sum_{k=1}^p \lambda_k = 1; \lambda_k \in \{0, 1\}, i = 1, 2, \dots, p. \tag{3.5}$$

Se $Q \in Z_+^{n \times p}$ a matriz em que cada coluna é um elemento de H , então as expressões (3.5) podem ainda ser reescritas como:

$$x = Q\lambda; e\lambda = 1; \lambda \in \{0, 1\}^{|H|}, \tag{3.6}$$

onde $e = (1, \dots, 1) \in \mathbb{R}^p$.

Substituindo essa expressão de x em (P), obtemos o problema mestre de Dantzig-Wolfe (MDW):

$$(MDW) \min (cQ)\lambda \quad (3.7a)$$

$$\text{s.a: } (AQ)\lambda = b \quad (3.7b)$$

$$e\lambda = 1 \quad (3.7c)$$

$$\lambda \in \{0, 1\}^{|H|} \quad (3.7d)$$

Chamemos a relaxação linear de MDW de RMDW. Mostra-se que RMDW é equivalente

$$\min\{cx : Ax = b, x \in \text{conv}(H)\}$$

Assim, a solução de RMDW fornece um limite inferior para (P) maior ou igual que aquele gerado pela relaxação linear de (P) [34].

No entanto, há uma dificuldade para a solução de (RMDW): p pode ser muito grande. Logo, teremos que usar a técnica de geração de colunas para a resolução do problema (ver subsecção 3.1.1). Estando resolvido o problema RMDW, restrito a algumas colunas de AQ , seja $\bar{\mu} \in R^{m_1}$ e $\bar{\nu} \in R$ os valores das variáveis duais associadas às restrições (3.7b) e (3.7c), respectivamente. Então, o menor custo reduzido, que vai gerar a próxima coluna a ser adicionada ao problema restrito, é dado pela solução do problema escravo a seguir:

$$(EDW) \delta = \min (c - \bar{\mu}A)x - \bar{\nu} \quad (3.8a)$$

$$\text{s.a: } Dx \leq d \quad (3.8b)$$

$$x \in Z_+^n \quad (3.8c)$$

Como no processo de geração de colunas, se $\delta \geq 0$ a solução do problema restrito corrente é ótima para RMDW. Do contrário, a coluna

$$\begin{bmatrix} cx^* \\ Ax^* \\ 1 \end{bmatrix},$$

onde x^* é a solução de (EDW), é introduzida em RMDW, e o processo é repetido.

É interessante escolhermos o particionamento das restrições de (P) de modo que a matriz D tenha uma boa estrutura, que permita resolver o problema escravo eficientemente. Por exemplo, se a matriz D é totalmente unimodular, as restrições de integralidade 3.8c podem ser descartadas, e o problema escravo torna-se um problema linear [34]. Por outro lado, neste caso $\text{conv}(H) = H$ e o valor da relaxação linear e de RMDW é o mesmo. Mostra-se portando que é preciso avaliar o compromisso entre a facilidade para resolver

(EDW) e a qualidade desejada do limite. Para uma melhor explanação sobre essa técnica, recomendamos uma consulta de [34].

O problema escravo (EDW) torna-se separável quando a matriz D tem uma estrutura bloco-diagonal, ou seja:

$$D = \begin{bmatrix} D_1 & & & & \\ & D_2 & & & \\ & & D_3 & & \\ & & & \ddots & \\ & & & & D_t \end{bmatrix}$$

onde $D_i \in R^{m_{2i} \times n_i}$, $\sum_{i=1}^t m_{2i} = m_2$, $\sum_{i=1}^t n_i = n$.

Considere a partição correspondente de $A = [A_1, A_2, \dots, A_t]$, com $A_i \in R^{m_1 \times n_i}$. Para cada $i = 1, 2, \dots, t$, defina $H_i = \{x_i \in Z_+^{n_i} : D_i x_i \leq d_i\}$ e seja Q_i a matriz cujas colunas são os elementos de H_i . Então

$$x_i \in H_i \iff x_i = Q_i \lambda_i, \quad e_i \lambda_i = 1, \quad \lambda_i \in \{0, 1\}^{|H_i|},$$

onde $e_i = (1, 1, \dots, 1) \in R^{|H_i|}$.

Observando que $H = H_1 \times H_2 \times \dots \times H_t$, o problema mestre (MDW) pode ser reescrito como o seguinte:

$$(MDW) \min \sum_{i=1}^t (c_i Q_i) \lambda_i \tag{3.9a}$$

$$\text{s a } \sum_{i=1}^t (A_i Q_i) \lambda_i = a \tag{3.9b}$$

$$e_i \lambda_i = 1, \quad \forall i = 1, 2, \dots, t \tag{3.9c}$$

$$\lambda_i \in \{0, 1\}^{|H_i|}, \quad \forall i = 1, 2, \dots, t \tag{3.9d}$$

Seja (\overline{MDW}) a relaxação linear do problema acima restrito a um subconjunto de colunas, ou seja, a um subconjunto de pontos dos H_i 's. Sejam $\bar{\mu}$ e $\bar{\nu}_i$ ($i = 1, 2, \dots, t$) a solução dual de (\overline{MDW}) . Os problemas escravos que fornecem novas colunas para o problema mestre são dados por:

$$(EDW_i) \quad \delta_i = \min (c_i - \bar{\mu} A_i) x_i - \nu_i \tag{3.10a}$$

$$\text{s.a.} \quad D_i x_i \leq d_i \tag{3.10b}$$

$$x_i \in Z_+^{n_i} \tag{3.10c}$$

para $i = 1, \dots, t$.

O menor custo reduzido é então:

$$\delta = \delta_i = \min_{i=1,2,\dots,t} \delta_i$$

Caso $\delta \geq 0$, a solução corrente é ótima para $(M\bar{D}W)$. Do contrário, para qualquer $i \in \{1, 2, \dots, t\}$ tal que $\delta_i < 0$, adiciona-se

$$\begin{bmatrix} c_i x_i^* \\ A_i x_i^* \\ I_i \end{bmatrix} \in R^{m_1+t+1}$$

onde x_i^* é a solução ótima de valor δ_i e I_i é a i -ésima coluna da identidade. Uma heurística usual é a escolha de $i = \hat{i}$ com $\delta = \delta_{\hat{i}} < 0$. Outra possibilidade é acrescentar várias colunas referentes a alguns índices i tais que $\delta_i < 0$.

3.2 Geração de Linhas

3.2.1 Princípio da Geração de Linhas

Consideramos agora o seguinte problema de programação linear (P') , que supomos ter muitas restrições ou mesmo que elas não sejam explicitamente conhecidas.

$$(P') \min cx \tag{3.11a}$$

$$\text{s.a: } Ax \leq b \tag{3.11b}$$

$$x \geq 0 \tag{3.11c}$$

Para resolver (P') , iniciamos resolvendo uma relaxação sua, dada, por exemplo, por um subconjunto de restrições de (3.11b). Então, se x^* é a solução obtida, consideramos o subproblema:

$$\gamma = \min b_i - A_i x^* \tag{3.12a}$$

$$\text{s.a: } (A_i, b_i) \in \text{lin}(A, b) \tag{3.12b}$$

onde $\text{lin}(A, b)$ representa o conjunto de linhas de (P') . Se $\gamma \geq 0$, a solução da relaxação é a solução de (P') . Do contrário, a restrição $A_{i^*} x \leq b_{i^*}$ é acrescentada, sendo (A_{i^*}, b_{i^*}) uma solução do subproblema (3.12b).

Um exemplo de uma formulação com muitas restrições é o seguinte modelo para determinar uma clique máxima de um grafo G . Associada a cada vértice $v \in V$, definimos

uma variável $\pi_v \in \{0, 1\}$ tal que $\pi_v = 1$ se, e somente se, v faz parte da clique máxima. Assim, a mencionada formulação é:

$$\text{(DCIM) max } \sum_{v \in V} \pi_v \quad (3.13a)$$

$$\text{s.a: } \sum_{v \in S} \pi_v \leq 1, \quad \forall S \subset \mathcal{S} \quad (3.13b)$$

$$\pi_v \in \{0, 1\}, \quad v \in V. \quad (3.13c)$$

A função objetivo (3.13a) procura escolher o menor número de vértices possível, considerando a condição estabelecida por (3.13b), de que no máximo um vértice de qualquer conjunto independente maximal será escolhido. Logo, a solução desse modelo é uma clique máxima. Vale observar que a relaxação linear de (3.13) é o dual da relaxação linear de (CIM).

Dados um subconjunto $\mathcal{S}' \subset \mathcal{S}$ e solução π^* da relaxação de (DCIM) definida por \mathcal{S}' , a geração de uma nova linha é dada pela solução do problema (3.3). Similarmente, se o valor de sua solução for maior que 1, a restrição definida por $S = \{v \in V | z_v = 1\}$ é acrescentada à relaxação corrente. Caso contrário, a solução da relaxação linear de (3.13) foi encontrada.

3.2.2 Métodos de Planos de Corte

Como vimos na subseção 2.2.3, uma estratégia usual para a resolução de um problema de programação inteira (PPI) consiste na adição sucessiva de restrições (ou linhas) à sua formulação, procurando aproximar o conjunto viável relaxado do poliedro associado. Uma dificuldade dessa abordagem reside no fato de que, em geral, não se conhece uma descrição explícita desse poliedro, que precisa, comumente, de um número muito grande de inequações. A alternativa é procurar descobrir iterativamente um subconjunto significativo de tais inequações.

Nesse processo de geração de linhas, vamos assumir que começamos resolvendo a relaxação linear de (PPI). Se sua solução ótima x^* satisfaz as condições de integralidade, então será também ótima para (PPI). Do contrário, é usual se considerar, em lugar do subproblema (3.12), o seguinte problema de separação:

$$\text{Encontre } (A_i, b_i) \in \text{lin}(A, b) \text{ tal que } A_i x^* > b_i \quad (3.14)$$

Assim, a restrição $A_i x \leq b_i$ encontrada “corta” o ponto x^* e outros pontos viáveis para a relaxação mas inviáveis para (PPI), sendo por isso chamada de plano de corte. Vale ressaltar que, dado a dificuldade de caracterizar $\text{lin}(A, b)$, os algoritmos de separação costumam ser específicos para cada problema e mesmo para cada classe de restrição de um mesmo problema.

3.3 Geração Simultânea de Linhas e Colunas

3.3.1 Princípio da Geração Simultânea

Como vimos nas seções anteriores, alguns problemas podem apresentar-se inviáveis de serem tratados com todas as suas variáveis (colunas) ou todas as suas restrições (linhas). Alguns desses problemas podem contemplar simultaneamente as duas dificuldades: muitas linhas e muitas colunas. Considere o seguinte problema P'' com muitas linhas e muitas colunas, onde possivelmente não se conhecem todas as linhas ou colunas inicialmente.

$$(P'') \min cx \tag{3.15a}$$

$$\text{s.a: } Ax \leq b \tag{3.15b}$$

$$x \geq 0 \tag{3.15c}$$

De início pode-se pensar em utilizar diretamente as duas técnicas anteriormente citadas. No entanto, ao gerarmos uma coluna com a técnica da subseção 3.1.1, estávamos utilizando implicitamente o fato de conhecermos todas as linhas, ou de pelo menos, estarmos trabalhando com um número fixo de restrições para as quais sabíamos o formato da coluna. O equivalente vale para a geração de linhas: ao determinarmos se alguma restrição é violada, consideramos todas (ou um conjunto fixo) de variáveis. Essa simplificação adotada nos métodos anteriores, e abandonada neste momento, permitia trabalharmos com subproblemas estáticos, que sempre responderiam às mesmas variáveis duais. Agora os subproblemas mudam para englobar as novas linhas ou colunas que foram geradas durante a execução do método.

Assim, o subproblema para a geração de linhas é

$$\gamma = \min_i b_i - A_i x^* \tag{3.16a}$$

$$\text{s.a: } (A_i, b_i) \in \text{lin}(A, b) \tag{3.16b}$$

onde $lin(A, b)$ representa o conjunto de linhas de (P'') restrito às colunas já geradas e x^* é a solução parcial obtida com essas colunas. Por outro lado, a geração de colunas é dada pelo subproblema

$$\delta = \min_j c_j - \mu A_j \quad (3.17a)$$

$$\text{s.a: } \begin{bmatrix} c_j \\ A_j \end{bmatrix} \in col(c, A) \quad (3.17b)$$

onde $col(c, A)$ representa o conjunto das colunas restrito às linhas geradas até o momento e μ são as variáveis duais das restrições já incluídas no modelo.

Caso $\gamma \geq 0$, a presente solução encontrada não viola qualquer restrição do problema (P'') restrito às colunas correntes. Por outro lado, caso $\delta \geq 0$, não há uma coluna que possa melhorar a função objetivo de (P'') , considerando apenas as restrições geradas até o momento.

3.3.2 Adicionando cortes à Decomposição Dantzig-Wolfe

Nesta seção, analisaremos o efeito da adição de um plano de corte ao problema de programação inteira (P), definido na seção 3.1.2, em sua decomposição de Dantzig-Wolfe. Duas são as possibilidades a serem consideradas: o corte é incorporado às restrições mantidas no mestre (3.7) ou o corte é transferido para o problema escravo (3.8). Estes casos são analisados separadamente a seguir, assumindo que $ax \leq \alpha$ é o corte a ser introduzido, onde $a \in \mathbb{R}^n \setminus \{0\}$ e $\alpha \in \mathbb{R}$. Relembremos que o problema (P), dado por:

$$(P) \min cx \quad (3.18a)$$

$$\text{s.a: } Ax = b \quad (3.18b)$$

$$Dx \leq d \quad (3.18c)$$

$$x \in Z_+^n \quad (3.18d)$$

é decomposto pela transferência de (3.18c) e (3.18d) para o problema escravo. Veremos como os problemas mestre e escravo são modificados com a adição do corte.

Cortes no Problema Mestre

Considerando $ax \leq \alpha$ acrescido às restrições (3.18b), o novo problema mestre será:

$$\min (cQ)\lambda \quad (3.19a)$$

$$\text{s.a: } (AQ)\lambda = b \quad (3.19b)$$

$$(aQ)\lambda \leq \alpha \quad (3.19c)$$

$$e\lambda = 1 \quad (3.19d)$$

$$\lambda \in \{0, 1\}^p \quad (3.19e)$$

Note que este problema pode ser obtido de (3.7) com a inclusão da restrição (3.19c). Isso daria origem a um novo problema escravo da seguinte forma:

$$\min (c - \bar{\mu}A)x - \bar{\nu} - \bar{\rho}ax \quad (3.20a)$$

$$\text{s.a: } Dx \leq d \quad (3.20b)$$

$$x \in Z_+^n \quad (3.20c)$$

onde $\bar{\rho}$ é o valor da variável dual associada à nova restrição. O novo problema escravo se diferencia do anterior pela inclusão da parcela $\bar{\rho}ax$ à função objetivo.

Cortes no Problema Escravo

Agora, assumiremos que o corte é adicionado às restrições que definem o problema escravo, ou mais precisamente, ao conjunto $H = \{x \in Z_+^n | Dx \leq d\}$. Neste caso o problema escravo é obtido diretamente de (3.8) com o acréscimo do corte, ou seja,

$$\delta = \min (c - \bar{\mu}A)x - \bar{\nu} \quad (3.21a)$$

$$\text{s.a: } Dx \leq d \quad (3.21b)$$

$$ax \leq \alpha \quad (3.21c)$$

$$x \in Z_+^n \quad (3.21d)$$

Por outro lado, em ambos os problemas mestres algumas variáveis λ_k são eliminadas. Observe que a estrutura original da matriz Q é modificada, conforme a adição do corte. Cada corte adicionado ao escravo elimina pontos do conjunto H , retirando colunas de Q . Isso é equivalente a fazer, em (3.7) $\lambda_k = 0$ para todo k tal que $aQ_k > \alpha$. Portanto, para a implementação dessa estratégia de inclusão dos cortes é necessária uma boa política de “limpeza” de colunas antigas, já geradas, que vão sendo eliminadas.

Capítulo 4

Formulação CIM via Formulação VR

4.1 Equivalência entre as Formulações

Nesta seção, mostraremos que as duas formulações do problema de coloração de vértices apresentadas no capítulo 2, quais sejam, formulação por conjuntos independentes maximais (CIM) e formulação por vértices representantes de cor (VR), estão fortemente relacionadas. Especificamente, mostraremos que a formulação (CIM) pode ser obtida a partir de uma decomposição, segundo Dantzig-Wolfe, da formulação (VR).

Para tal, recapitulemos a versão assimétrica da formulação por representantes de cor, apresentada na subseção 2.3.2:

$$\text{(VRA) min } \sum_{v \in V} x_{vv} \quad (4.1a)$$

$$\text{s.a: } \sum_{u \in \bar{N}^-[v]} x_{uv} \geq 1, \quad \forall v \in V, \quad (4.1b)$$

$$x \in \Omega \quad (4.1c)$$

onde

$$\Omega = \{x \in \mathbb{B}^{n+\bar{m}} : x_{uv} + x_{uw} \leq x_{uu}, \forall u \in V, \forall vw \in E[\bar{N}^+(u)], \\ x_{uv} \leq x_{uu}, \forall u \in V, \forall v \text{ isolado em } \bar{N}^+(u) \}.$$

Primeiramente, observe que a matriz associada às restrições lineares que definem Ω apresenta uma estrutura bloco-angular. De fato, para cada $u \in V$ fixo, o grupo de restrições

$$x_{uv} + x_{uw} \leq x_{uu}, \forall vw \in E[\tilde{N}^+(u)], \quad (4.2a)$$

$$x_{uv} \leq x_{uu}, \quad \forall v \text{ isolado em } \tilde{N}^+(u) \quad (4.2b)$$

envolve apenas as variáveis x_{uv} , para $v \in \tilde{N}^+[u]$. Denotando tais variáveis por x_u e definindo

$$\Omega_u = \{x_u \in \mathbb{B}^{[N^+[u]]} : x_u \text{ satisfaz (4.2)}\},$$

temos que Ω é o produto cartesiano dos conjuntos Ω_u , para $u \in V$, ou seja,

$$\Omega = \prod_{u \in V} \Omega_u,$$

Note ainda que $x_u \in \Omega_u$ se, e somente se, x_u corresponde ao vetor característico de um conjunto independente em $G[N^+[u]]$ contendo u ou $x_u = 0$ (o vetor característico de um “conjunto independente” vazio). Consequentemente, dados $x \in \Omega$, $u \in V$ e $v \in N^+[u]$, temos que

$$x_{uv} = \sum_{S \in \mathcal{S}^+(u) \cup \{\emptyset\}} \lambda_S x_{uv}^S, \quad \sum_{S \in \mathcal{S}^+(u) \cup \{\emptyset\}} \lambda_S = 1, \quad \lambda_S \in \mathbb{B}, \forall S \in \mathcal{S}^+(u) \cup \{\emptyset\}, \quad (4.3)$$

onde $\mathcal{S}^+(u)$ representa o conjunto de conjuntos independentes de G contendo u e vértices em $N^+(u)$, e x^S é o vetor característico do conjunto independente $S \in \mathcal{S}^+(u) \cup \{\emptyset\}$. Uma vez que $\mathcal{S}^+(u) \neq \emptyset$ e $x_{uv}^\emptyset = 0$, as expressões em (4.3) podem ser simplificadas como

$$x_{uv} = \sum_{S \in \mathcal{S}^+(u): v \in S} \lambda_S, \quad \sum_{S \in \mathcal{S}^+(u)} \lambda_S \leq 1, \quad \lambda_S \in \mathbb{B}, \forall S \in \mathcal{S}^+(u). \quad (4.4)$$

Particularmente para $u = v$, a expressão de x_{uv} torna-se

$$x_{uu} = \sum_{S \in \mathcal{S}^+(u)} \lambda_S. \quad (4.5)$$

Convém observar que, a rigor, deveríamos indexar as variáveis λ_S em (4.3)-(4.5) também com respeito ao vértice u a que se referem. Entretanto, como $\mathcal{S}^+(u) \cap \mathcal{S}^+(u') = \emptyset$, para $u \neq u'$, tal diferenciação é desnecessária.

A partir da observação anterior, vemos que uma decomposição da formulação (4.1), segundo Dantzig-Wolfe, gera o seguinte problema mestre

$$\text{(MRC) min } \sum_{u \in V} \sum_{S \in \mathcal{S}^+(u)} \lambda_S \quad (4.6a)$$

$$\text{s.a: } \sum_{u \in \bar{N}^-[v]} \sum_{S \in \mathcal{S}^+(u): v \in S} \lambda_S \geq 1, \quad \forall v \in V, \quad (4.6b)$$

$$\sum_{S \in \mathcal{S}^+(u)} \lambda_S \leq 1, \quad \forall u \in V, \quad (4.6c)$$

$$\lambda_S \in \mathbb{B}, \quad \forall u \in V, \forall S \in \mathcal{S}^+(u), \quad (4.6d)$$

e o seguinte problema escravo, para cada $u \in V$,

$$\delta(u) = \min \left\{ - \sum_{v \in N^+[u]} \bar{\pi}_v x_{uv} - \bar{\nu}_u + 1 : x_u \in \Omega_u, x_{uu} = 1 \right\}, \quad (4.7)$$

onde $\bar{\pi}_v$ e $\bar{\nu}_u$ são os valores duais associados, respectivamente, às restrições (4.6b)-(4.6c) do subproblema restrito corrente. Vale destacar que a fixação $x_{uu} = 1$ em (4.7) decorre do fato de termos eliminado a variável λ_\emptyset associada a u , restando, por conseguinte, apenas as variáveis indexadas por conjuntos independentes contendo u .

Desta forma, para cada $u \in V$, o menor custo reduzido de uma variável λ_S , com $S \in \mathcal{S}^+(u)$, é dado por

$$\delta(u) = 1 - \bar{\pi}_u - \bar{\nu}_u - \delta'(u),$$

onde

$$\delta'(u) = \max \sum_{v \in N^+(u)} \bar{\pi}_v x_{uv} \quad (4.8a)$$

$$\text{s.a: } x_{uv} + x_{uw} \leq 1, \quad \forall vw \in E[\bar{N}^+(u)], \quad (4.8b)$$

$$x_{uv} \in \mathbb{B}, \quad \forall v \in \bar{N}^+(u). \quad (4.8c)$$

Usando o fato que $\bar{\pi}_v \geq 0$, para todo $v \in V$, podemos concluir que o problema acima sempre possui solução que caracteriza um conjunto independente maximal em $G[N^+(u)]$. Sendo assim, podemos restringir as colunas da formulação (4.6), redefinindo o conjunto $\mathcal{S}^+(u)$ como

$$\mathcal{S}^+(u) = \{S \in \bar{\mathcal{S}} : u \in S \subset N^+[u], S \text{ é maximal em } G[N^+[u]]\}.$$

Com isso, as variáveis de (4.6) estão indexadas pelo conjunto

$$\mathcal{S}^+ = \bigcup_{u \in V} \mathcal{S}^+(u).$$

Adicionalmente, a formulação (4.6) pode ser resescrita como

$$\text{(MRC) min } \sum_{S \in \mathcal{S}^+} \lambda_S \quad (4.9a)$$

$$\text{s.a: } \sum_{S \in \mathcal{S}^+ : v \in S} \lambda_S \geq 1, \quad \forall v \in V, \quad (4.9b)$$

$$\sum_{S \in \mathcal{S}^+(u)} \lambda_S \leq 1, \quad \forall u \in V, \quad (4.9c)$$

$$\lambda_S \in \mathbb{B}, \quad \forall S \in \mathcal{S}^+, \quad (4.9d)$$

Note que as funções objetivo (4.6a) e (4.9a) são equivalentes uma vez que $\mathcal{S}^+(u) \cap \mathcal{S}^+(u') = \emptyset$, para $u \neq u'$. Uma equivalência ocorre também entre as restrições (4.6b) e (4.9b) devido a igualdade entre os conjuntos $\{S \in \bar{\mathcal{S}} : v \in S \text{ e } S \in \mathcal{S}^+(u) \text{ para algum } u \in \bar{N}^-[v]\}$ e $\{S \in \mathcal{S}^+ : v \in S\}$. De fato, o primeiro conjunto está claramente contido no segundo e, dado $S \in \mathcal{S}^+$ tal que $v \in S$, conclui-se que $S \in \mathcal{S}^+(u)$ com u sendo o menor vértice em S .

No que se segue, vamos analisar semelhanças e diferenças entre o problema mestre (4.9) e a formulação por conjuntos independentes maximais,

$$\text{(CIM) min } \sum_{S \in \mathcal{S}} \lambda_S \quad (4.10a)$$

$$\text{s.a: } \sum_{S \in \mathcal{S} : v \in S} \lambda_S \geq 1, \quad \forall v \in V, \quad (4.10b)$$

$$\lambda_S \in \mathbb{B}, \quad \forall S \in \mathcal{S} \quad (4.10c)$$

bem com entre o conjunto de subproblemas escravos (4.8) e o problema do conjunto independente máximo ponderado

$$\delta' = \max \sum_{v \in V} \bar{\pi}_v z_v \quad (4.11a)$$

$$\text{s.a: } z_v + z_u \leq 1, \quad \forall vw \in E, \quad (4.11b)$$

$$z_v \in \mathbb{B}, \quad \forall v \in V. \quad (4.11c)$$

Lembramos que o custo reduzido da coluna gerada pela solução de (4.11) é

$$\delta = 1 - \delta'.$$

A primeira constatação é que (4.9) possui mais colunas que (4.10), como estabelece o seguinte resultado.

Proposição 6. $\mathcal{S} \subseteq \mathcal{S}^+$.

Prova: Seja $S \in \mathcal{S}$ e considere o menor vértice u em S . Então, $S \in \mathcal{S}^+(u) \subseteq \mathcal{S}^+$. \square

Além disso, é possível que toda solução ótima de (4.9) tenha uma variável não-nula em $\mathcal{S}^+ \setminus \mathcal{S}$. De fato, no grafo da figura 4.1, todo conjunto independente maximal contém exatamente um vértice u de $\{1, 2, 3\}$ e pertence a $\mathcal{S}^+(u)$. Como são necessárias quatro cores para colorir o grafo, em qualquer solução ótima de (4.10), um vértice u irá representar pelo menos dois conjuntos em $\mathcal{S}^+(u)$. Logo, uma das restrições (4.9c) não é satisfeita. Consequentemente, qualquer solução ótima de (4.9) usará um conjunto de $\mathcal{S}^+ \setminus \mathcal{S}$.

Figura 4.1: Exemplo que mostra a diferença entre as formulações CIM e MRC

Mesmo para o grafo da figura 4.1, entretanto, tal situação pode não acontecer se outra ordenação dos vértices fosse considerada, por exemplo, se os vértices em $\{4, 5, 6, 7\}$ recebessem os menores rótulos. Nesse caso, ambas as formulações compartilhariam uma solução ótima.

Mostramos a seguir que, para uma dada ordenação dos vértices, são exatamente as restrições (4.9c) que podem levar ao caso em que seja vazia a interseção entre os conjuntos de colorações ótimas apontados pelas duas formulações. Observe por outro lado que a exclusão das restrições (4.9c) não altera o valor ótimo de (4.9).

Proposição 7. *Na ausência das restrições (4.9c), toda solução ótima de (4.10) é solução ótima de (4.9).*

Prova: Mesmo na ausência de (4.9c), o valor ótimo de (4.9) é igual àquele da formulação

(4.10). Além disso, pela Proposição 6, toda solução ótima de (4.10) pode ser estendida a uma solução viável de (4.9b) e (4.9d), fazendo $\lambda_S = 0$, para todo $S \in \mathcal{S}^+ \setminus \mathcal{S}$. Logo, o resultado segue. \square

Com respeito aos problemas que geram as colunas das duas formulações, podemos fazer as seguintes observações. Como ilustra o exemplo da figura 4.1, existem casos em que os conjuntos das colunas geradas precisam ser necessariamente diferentes, implicando dizer que o conjunto de subproblemas (4.8) e o problema do conjunto independente máximo ponderado (4.11) não são equivalentes. Entretanto, a seguinte equivalência ocorre quando as restrições (4.9c) são desconsideradas.

Proposição 8. *Na ausência das restrições (4.9c) e considerando o mesmo vetor de valores duais $\bar{\pi}$, é válido que:*

$$(i) \quad \delta = \min\{\delta(u) : u \in V\}$$

(ii) *Todo conjunto independente $S^* \in \mathcal{S}$ que é solução do problema (4.11) é também solução do problema (4.8), para algum $u^* \in \arg \min\{\delta(u) : u \in V\}$.*

Prova: Como $\bar{v} = 0$, pois as restrições (4.9c) foram excluídas, o custo de uma coluna indexada por $S \in \mathcal{S}$ é o mesmo nas duas formulações. Adicionalmente, sendo $\bar{\pi} \geq 0$, o custo de uma coluna indexada por $S \in \mathcal{S}^+ \setminus \mathcal{S}$ é sempre dominado por aquele de uma coluna em \mathcal{S} . Isto mostra a primeira parte. Adicionalmente, se o conjunto independente S^* define uma solução ótima de (4.11), então o menor vértice u^* de S^* pertence a $\arg \min\{\delta(u) : u \in V\}$. Assim, mostra-se a segunda parte. \square

As Proposições 7 e 8 mostram que, se as restrições (4.9c) são desconsideradas, os processos de geração de colunas determinados pela formulação por conjuntos independentes maximais e pela decomposição de Dantzig-Wolfe da formulação por representantes de cor são equivalentes. Isto implica afirmar que as variáveis indexadas por $\mathcal{S}^+ \setminus \mathcal{S}$ poderiam ser fixadas em zero, ou seja, eliminadas do modelo.

Neste ponto, gostaríamos de reforçar que uma possível diferença entre as formulações (4.9) e (4.10) seria decorrente das restrições (4.9c) e da ordenação dada aos vértices. Entretanto, não é claro que essa diferença possa se refletir na qualidade do limite inferior gerado pelas relaxações lineares das duas formulações. Sendo assim, optamos por aprofundar o estudo com a formulação (4.10), para a qual podemos encontrar alguns resultados com respeito à estrutura do politopo associado. Além disso, os resultados dos

experimentos irão independer de uma ordenação escolhida a priori.

4.2 Desigualdades Válidas para CIM

Em [5], Campêlo, Corrêa e Frota apresentam duas classes de facetas para a formulação por representantes de cor. Ambas são definidas por subestruturas do grafo, que incluem cliques, buracos e anti-buracos. A primeira classe considera o tamanho de um maior conjunto independente da estrutura (subgrafo), enquanto a segunda observa o seu número cromático. Adaptações dessas classes de facetas para a versão assimétrica da formulação foram em seguida apresentadas por Campêlo, Campos e Corrêa [4].

Nesta seção, utilizamos a segunda classe de facetas e a transformação (4.4) para derivarmos desigualdades válidas para a formulação por conjuntos independentes. Em seguida, estabelecemos condições sob as quais elas também são definidoras de facetas. Destacamos que desenvolvimento similar aplicado à primeira classe de facetas conduz apenas a desigualdades redundantes para a formulação (CIM) e, por isso, não será aqui apresentado.

Por simplicidade, vamos usar como base as desigualdades válidas expressas para formulação original (VR). Adiantamos que os mesmos resultados seriam obtidos com as versões correspondentes para a formulação assimétrica. Tais desigualdades são caracterizadas em [5] como seguir.

Teorema 6. *Se $H \subseteq V$, $H \neq \emptyset$, então*

$$\sum_{v \in H} x_{vv} + \sum_{v \in H} \sum_{u \in \bar{N}(v) \setminus H} x_{uv} \geq \chi(G[H]) \quad (4.12)$$

é uma desigualdade válida para o politopo associado a (VR). Adicionalmente, ela define uma faceta se $G[H]$ é $\chi(G[H])$ -crítico e $\bar{G}[H]$ é conexo.

Conforme observado Campêlo, Correa e Frota [5], se $H \subseteq V$ induz um buraco ou anti-buraco de G , então a desigualdade (4.12) correspondente define uma faceta. O mesmo não ocorre, entretanto, se $G[H]$ é uma clique, outra estrutura χ -crítica fortemente ligada ao problema de coloração, tendo em vista que $\bar{G}[H]$ é desconexo. Por outro lado, mostramos a seguir que cliques também podem ser usadas em conjunto com o Teorema 6 de uma maneira indireta. Na verdade, estabelecemos no Corolário 3 uma forma para encontrar

outras estruturas do grafo que também induzam facetas, usando como base subgrafos críticos não necessariamente conexos. Conforme veremos no capítulo 6, o uso dessas facetas mostrou-se bastante eficiente em várias instâncias.

Corolário 3. *Seja $H \subseteq V$, $H \neq \emptyset$. Se existe $H' \subset H$ tal que $G[H']$ é $\chi(G[H'])$ -crítico e $G[H]$ é obtido a partir de $G[H']$ por uma seqüência (não vazia) de operações de Mycielski, então a desigualdade (4.12) define uma faceta do politopo associado a (VR).*

Prova: Pela Proposição (3), concluímos que $G[H]$ é $\chi(G[H])$ -crítico, posto que $G[H']$ é $\chi(G[H'])$ -crítico. Segundo o Teorema 6, resta mostrar que $\bar{G}[H]$ é conexo. Na verdade, é suficiente mostrar que, para qualquer grafo $F = (U, A)$, o complemento de $M(F) = (U \cup W \cup \{z\}, A')$ é um grafo conexo. Primeiro observe que, em $\overline{M(F)}$, W define uma clique, todo vértice em W se liga a algum vértice em U e que todo vértice em U está conectado a z . Então, tome dois vértices u, w de $M(F)$ e considere dois casos. No primeiro, assuma que $u \in W$. Assim, sempre existe caminho ligando u a w em $\overline{M(F)}$: diretamente, se $w \in W$; usando um vértice de U , se $w = z$; ou usando um vértice de U e z , se $w \in U$. No segundo caso, assuma que $u \in U$. Também agora existe caminho direto, para $w = z$, ou usando z como intermediário, para $w \in U$. Estes dois casos mostram que $\overline{M(F)}$ é conexo. \square

Agora, vamos adaptar a desigualdade (4.12) às formulações (MRC) e (CIM). Usando (4.4)-(4.5), derivamos abaixo a desigualdade correspondente expressa em função de λ .

$$\begin{aligned}
\chi(G[H]) &\leq \sum_{v \in H} x_{vv} + \sum_{v \in H} \sum_{u \in \bar{N}(v) \setminus H} x_{uv} \\
&= \sum_{u \in H} x_{uu} + \sum_{u \in V \setminus H} \sum_{v \in \bar{N}(u) \cap H} x_{uv} \\
&= \sum_{u \in H} \sum_{S \in \mathcal{S}^+(u)} \lambda_S + \sum_{u \in V \setminus H} \sum_{v \in \bar{N}(u) \cap H} \sum_{S \in \mathcal{S}^+(u): v \in S} \lambda_S \\
&= \sum_{u \in H} \sum_{S \in \mathcal{S}^+(u)} \lambda_S + \sum_{u \in V \setminus H} \sum_{S \in \mathcal{S}^+(u)} \sum_{v \in S \cap H} \lambda_S \\
&= \sum_{u \in H} \sum_{S \in \mathcal{S}^+(u)} \lambda_S + \sum_{u \in V \setminus H} \sum_{S \in \mathcal{S}^+(u)} |S \cap H| \lambda_S
\end{aligned}$$

Essa desigualdade não define faceta do politopo associado a (MRC). De fato, reduzindo os coeficientes $|S \cap H|$ para 1, obtemos a desigualdade dominante

$$\sum_{u \in V} \sum_{S \in \mathcal{S}^+(u): S \cap H \neq \emptyset} \lambda_S \geq \chi(G[H]), \tag{4.13}$$

que é também válida, conforme demonstra a proposição abaixo. Observe que as expressões (4.13) e (4.14) são iguais.

Proposição 9. *Se $H \subseteq V$, $H \neq \emptyset$, então a desigualdade*

$$\sum_{S \in \mathcal{S}^+ : S \cap H \neq \emptyset} \lambda_S \geq \chi(G[H]) \quad (4.14)$$

é válida para o politopo associado a (MRC).

Prova: Em toda coloração de G , pelo menos $\chi(G[H])$ cores são necessárias para colorir $G[H]$. Logo, pelo menos $\chi(G[H])$ conjuntos independentes de \mathcal{S}^+ que interceptam H devem ser escolhidos. \square

Com respeito à formulação por conjuntos independentes maximais (CIM), dada a relação com (MRC), chegamos diretamente ao resultado abaixo. No que se segue, vamos denotar por $P(\mathcal{S})$ o politopo associado a (CIM), isto é,

$$P(\mathcal{S}) = \text{conv}\{\lambda \in \mathbb{B}^{|\mathcal{S}|} : \lambda \text{ satisfaz (4.10b)}\}.$$

Proposição 10. *Se $H \subseteq V$, $H \neq \emptyset$, então a desigualdade*

$$\sum_{S \in \mathcal{S}_H} \lambda_S \geq \chi(G[H]), \quad (4.15)$$

onde

$$\mathcal{S}_H = \{S \in \mathcal{S} : S \cap H \neq \emptyset\}, \quad (4.16)$$

é válida para o politopo $P(\mathcal{S})$.

Na proposição acima, vale a pena destacar os casos extremos para a escolha de H , quais sejam, um único vértice ou todos eles. Se $H = \{v\}$, a desigualdade (4.15) torna-se a restrição (4.10b). No outro extremo, $H = V$, o primeiro termo de (4.15) é a função objetivo, que soma todas as variáveis do modelo.

Para estabelecermos condições sob as quais a desigualdade (4.15) define faceta para $P(\mathcal{S})$, vamos nos basear em alguns resultados conhecidos sobre a estrutura facial do politopo associado ao problema de cobertura por conjuntos, definido por

$$P_C(A) = \text{conv}\{x \in \mathbb{B}^q : Ax \geq 1\},$$

onde $A \in \mathbb{B}^{p \times q}$ é uma matriz com linhas indexadas por $I = \{1, 2, \dots, p\}$ e colunas indexadas por $J = \{1, 2, \dots, q\}$. Observe que $P(\mathcal{S})$ é um politopo de cobertura de conjuntos,

onde

$$I = V, \quad J = \mathcal{S} \quad \text{e} \quad A = [a_{vS}] \in \mathbb{B}^{n \times |\mathcal{S}|}, \quad \text{com } a_{vS} = 1 \text{ se, e somente se, } v \in S. \quad (4.17)$$

Preliminarmente, vamos apresentar algumas definições associadas a $P_C(A)$, introduzidas por Cornuéjols e Sassano [12] e Sassano [33]. O grafo de incidência de A é o grafo bipartite $B = (I, J, E)$, tal que $(i, j) \in E$ se, e somente se, $a_{ij} = 1$. Considere um conjunto $I' \subseteq I$. Um conjunto $J' \subseteq J$ é uma cobertura de I' se todo vértice $i \in I'$ é adjacente a pelo menos um vértice em J' . A cardinalidade da menor cobertura de I' é denotada por $\beta(I')$. Observe que existe uma correspondência biunívoca entre os pontos de $\{x \in B^q : Ax \geq 1\}$ e as coberturas de I . O valor $\beta(I)$ é chamado o número de cobertura de A . Adicionalmente, é definido o grafo $G^* = (J, E^*)$, onde $E^* = \{(j, k) : \beta(I \setminus (N_B(j) \cap N_B(k))) = \beta(I) - 1\}$. Note que $\beta(I \setminus (N_B(j) \cap N_B(k))) \in \{\beta(I) - 1, \beta(I)\}$, pois $N_B(j) \cap N_B(k) \subset N_B(j)$ [33].

O politopo $P_C(A)$ é não-vazio se, e somente se, $|N_B(i)| \geq 1$ para todo $i \in I$. Além disso, Sassano [33] apresenta os seguintes resultados.

Proposição 11. *O politopo $P_C(A)$ tem dimensão cheia se, e somente se, $|N_B(i)| \geq 2$ para todo $i \in I$.*

Proposição 12. *Assuma que $P_C(A)$ tem dimensão cheia. Então as seguintes afirmativas são verdadeiras:*

- (i) *Para cada $j \in J$, $x_j \leq 1$ define uma faceta de $P_C(A)$;*
- (ii) *Para cada $j \in J$, $x_j \geq 0$ define uma faceta de $P_C(A)$ se, e somente se, $|N_B(i) \setminus \{j\}| \geq 2$, para todo $i \in I$;*
- (iii) *A desigualdade*

$$\sum_{j \in J} x_j \geq \beta(I) \quad (4.18)$$

define uma faceta de $P_C(A)$ se G^ é conexo.*

Dado $J' \subset J$, $J \neq J'$, vamos definir

$$I(J') = \{i \in I : a_{ij} = 0 \forall j \in J \setminus J'\} = \{i \in I : N_B(i) \subseteq J'\}.$$

e denotar por $A(J')$ a submatriz de A com colunas indexadas por J' e linhas indexadas por $I(J')$. Note que $A(J')$ define um problema de cobertura e que seu número de cobertura é $\beta(I(J'))$. Cornuéjols e Sassano [12] obtêm a seguinte propriedade.

Proposição 13. *Seja $J' \subset J$, $J \neq J'$, e assumamos que*

$$\sum_{j \in J'} x_j \geq \beta(I(J')) \quad (4.19)$$

defina uma faceta de $P_C(A(J'))$ e $P_C(A)$ tem dimensão cheia. Então (4.19) define uma faceta de $P_C(A)$ se, e somente se, $\beta(I(J' \cup \{j\})) = \beta(I(J'))$ para todo $j \in J \setminus J'$.

Usando (4.17), podemos particularizar a exposição acima para o politopo $P(\mathcal{S})$. Para este politopo, temos que

$$B = (V, \mathcal{S}, \{(v, S) : v \in S\}).$$

Assim, para $v \in V$, $N_B(v)$ é o conjunto de conjuntos independentes maximais de G que contêm v , ou seja,

$$N_B(v) = \mathcal{S}_{\{v\}} = \{S \in \mathcal{S} : v \in S\}, \forall v \in V.$$

Por outro lado,

$$N_B(S) = S, \forall S \in \mathcal{S}.$$

O valor $\beta(H)$, para $H \subseteq V$, é o menor número de conjuntos independentes de G que cobrem H , o que resulta em

$$\beta(H) = \chi(G[H]).$$

Adicionalmente,

$$G^* = (\mathcal{S}, \{(S, S') : \chi(G[V \setminus (S \cap S')]) = \chi(G) - 1\}). \quad (4.20)$$

Em nosso contexto, é conveniente reescrevermos as proposições 12 e 13 como segue:

Proposição 14. *Seja $H \subseteq V$, $H \neq \emptyset$ e assumindo que*

$$\sum \lambda_{S \in \mathcal{S}_H} \geq \chi(G[V(S_H)]) \quad (4.21)$$

defina faceta de $P(S_H)$. Então (4.21) define faceta de $P(S)$ se, e somente se, $\chi(G[V(S_H \cup \{S\}]]) = \chi(G[V(S_H)])$, para todo $S \in \mathcal{S} \setminus \mathcal{S}_H$.

Proposição 15. *Seja $H \subseteq V$, $H \neq \emptyset$. A desigualdade*

$$\sum a_S \lambda_{S \in \mathcal{S}_H} \geq a_0 \quad (4.22)$$

é válida para $P(S)$ se, e somente se, é válida para $P(S_H)$

Além disso, os seguintes resultados auxiliares são úteis.

Proposição 16. *Dado $H \subseteq V$, $H \neq \emptyset$, as seguintes assertivas são verdadeiras.*

$$(i) \quad H \subseteq V(S_H) = V \setminus \bigcup_{S \in S_H} S$$

$$(ii) \quad S_H = S_{V(S_H)}$$

$$(iii) \quad \text{Para todo } S \in \mathcal{S}_{\mathcal{H}}, V(S_H \cup \{S\}) \subseteq V(S_H) \cup S$$

Prova:

(i) *Seja $v \in H$. Então, para todo $S \subseteq S_{\{v\}}$, $S \cap H \supseteq \{v\}$. Logo, $S_{\{v\}} \subseteq S_H = \{S \in \mathcal{S} : S \cap H \neq \emptyset\}$. É válido que $v \in V(S_H)$. Assim, mostramos a inclusão $H \subseteq V(S_H)$. Já a igualdade decorre das seguintes equivalências. Para todo $v \in V$:*

$$v \in V(S_H) \Leftrightarrow S_{\{v\}} \subseteq S_H \Leftrightarrow (S \in \mathcal{S}, v \in S \Rightarrow S \in S_H) \Leftrightarrow (S \in \mathcal{S} \setminus S_H \Rightarrow v \notin S) \Leftrightarrow v \notin \bigcup_{S \in \mathcal{S} \setminus S_H} S.$$

(ii) *De (i) segue diretamente que $S_H \subseteq S_{V(S_H)}$. Por outro lado, seja $S \in S_{V(S_H)}$. Tome $v \in S \cap V(S_H) \neq \emptyset$. Então, $S \in S_{\{v\}} \subseteq S_H$.*

(iii) *Sejam $S \in \mathcal{S} \setminus S_H$ e $v \in V(S_H \cup \{S\})$. Se $v \notin S$, então $S_{\{v\}} \subseteq S_H \cup \{S\}$ resulta em $S_{\{v\}} \subseteq S_H$, ou seja, $v \in V(S_H)$. Logo, $v \in S \cup V(S_H)$.*

□

O grafo da figura 4.3 mostra que inclusões estritas podem ocorrer nos itens (i) e (iii) da Proposição 16. Primeiro, considerando $H = \{1, 2, 3, 4, 5, 6, 7\}$ temos que todo conjunto independente de G intercepta H , ou seja, $S_H = \mathcal{S}$. Por conseguinte, $H \neq V(S_H) = V$. Agora, tomando $H = \{4, 6, 8, 9\}$ passamos a ter $\mathcal{S} \setminus S_H = \{S_1, S_2\}$, onde $S_1 = \{1, 3, 5\}$ e $S_2 = \{2, 5, 7\}$. Neste caso, $V(S_H) = H$, $V(S_H \cup S_1) = V \setminus S_2 \neq H \cup S_1 = V(S_H) \cup S_1$ e, simultaneamente, $V(S_H \cup S_2) \neq V(S_2) \cup S_2$.

A partir da Proposição 11 podemos determinar a dimensão de $P(\mathcal{S})$.

Corolário 4. *$P(\mathcal{S})$ tem dimensão cheia.*

Prova: Seja $v \in V$. Devemos mostrar que $|N_B(v)| \geq 2$, ou seja, que v pertence a pelo menos dois conjuntos independentes maximais de G . Suponha que v participa de um único $S \in \mathcal{S}$. Logo, $N_G(v) = V \setminus S$. Se $S = \{v\}$, então v é um vértice universal. Do

contrário, $u \in S \setminus \{v\}$ é dominado por v . Como ambos os casos são impossíveis pelas hipóteses H1 e H2, concluímos que $|N_B(v)| \geq 2$. \square

Já a Proposição 12 estabelece em que casos as desigualdades triviais definem facetas.

Corolário 5. *Se $S \in \mathcal{S}$. A desigualdade $\lambda_S \leq 1$ define faceta de $P(\mathcal{S})$, acontecendo o mesmo para $\lambda_S \geq 0$ se, e somente se, para todo $v \in V$, existem pelo menos dois conjuntos independentes em $\mathcal{S} \setminus \{S\}$ contendo v .*

Com respeito às desigualdades (4.15), considere primeiro o caso em que elas se reduzem às restrições (4.9b). Para este caso, Hansen, Labbé e Schindl [20] usam a Proposição 13 para mostram que (4.9b) definem facetas se v não é dominado. Em nosso contexto, considerando as hipóteses H1 e H2, enunciamos o resultado como a seguir.

Proposição 17. *Para $H = \{v\}$, $v \in V$, a desigualdade (4.15) define uma faceta de $P(\mathcal{S})$.*

Também para o outro caso extremo, mencionado anteriormente, em que H compreende todos os vértices, Hansen, Labbé e Schindl [20] obtêm a propriedade abaixo a partir da Proposição 12(iii).

Proposição 18. *Assuma que G é $\chi(G)$ -crítico. Então a desigualdade (4.15) para $H = V$ define uma faceta de $P(\mathcal{S})$ se, e somente se, \bar{G} é conexo.*

Na verdade, o resultado da Proposição 18 pode ser ligeiramente estendido, usando argumentação bastante similar àquela usada por Hansen, Labbé e Schindl [20] em sua demonstração, conforme a seguir.

Proposição 19. *Seja $H \subset V$, $H \neq \emptyset$. Se $G[H]$ é um subgrafo $\chi(G[V(S_H)])$ -crítico, então a desigualdade (4.15) define uma faceta de $P(S_H)$. Se $\bar{G}[H]$ é desconexo, então (4.15) não define faceta de $P(S_H)$ nem de $P(\mathcal{S})$.*

Prova:

- Agora, assumamos que $\bar{G}[H]$ não é conexo. Então, seja $H' \subset H$ tal que $\bar{G}[H']$ não se conecta a $\bar{G}[H \setminus H']$ ou, equivalentemente, tal que todos os vértices de $G[H']$ se ligam a todos os vértices de $\bar{G}[H \setminus H']$. Então, $\chi(G[H']) + \chi(G[H \setminus H'])$ e podemos particionar \mathcal{S}_H em dois subconjuntos disjuntos $\mathcal{S}_{H'}$ e $\mathcal{S}_{H \setminus H'}$. Assim, (4.15) resulta

da soma das desigualdades

$$\begin{aligned}\sum_{S \in \mathcal{S}_{H'}} \lambda_S &\geq \chi(G[H']), \\ \sum_{S \in \mathcal{S}_{H \setminus H'}} \lambda_S &\geq \chi(G[H \setminus H']).\end{aligned}$$

E como estas desigualdades são válidas para $P(S_H)$, pelas Preposições 10 e 15, concluímos que (4.15) não define faceta para $P(S_H)$. A mesma conclusão estende-se a $P(\mathcal{S})$, pela preposição 15.

- Assuma que $G[H]$ é $\chi(G[V(S_H)])$ -crítico e $\bar{G}[H]$ é conexo. Usando o Corolário 4, conclui-se que $P(S_H)$ tem dimensão cheia. Então considerando que $\chi(G[V(S_H)]) = \chi(G[H])$ e usando a Proposição 12(iii) resta mostrar que $G^*[S_H]$ é conexo. Observe que $E(G^*)$ é igual a

$$E^* = \{(S, S') \in S_H \times S_H : S \cap S' \neq \emptyset\}.$$

Por um lado, a inclusão $E(G^*) \subseteq E^*$ decorre da definição de $E(G^*)$ em (4.20); por outro, sendo $G[H]$ crítico, temos que $\chi(G[H \setminus (S \cap S')]) = \chi(G[H]) - 1$ quando $(S, S') \in E^*$, o que mostra a inclusão inversa. Já a conexidade de G^* é garantida pela hipótese de que $\bar{G}[H]$ é conexo. De fato, sob esta hipótese, podemos dizer que, entre cada par de vértices v e w de \bar{G} , existe uma seqüência de cliques maximais K_1, K_2, \dots, K_p em \bar{G} tais que $K_i \cap K_{i+1} \neq \emptyset$, para todo $i = 1, 2, \dots, k - 1$. Isto equivale afirmar que existe em G uma seqüência de conjuntos independentes maximais com a mesma propriedade. Tal fato e a expressão de E^* mostram a conexidade de G^* . Concluímos assim que (4.15) define uma faceta para $P(S_H)$.

□

Corolário 6. *Seja $H \subseteq V$. Se $G[H]$ é $\chi(G)$ -crítico e $\bar{G}[H]$ é conexo, então, (4.15) define faceta de $P(\mathcal{S})$. Prova: Como consequência da Proposição 16(i), $\chi(G[H]) = \chi(G[V(S_H)]) = \chi(G)$. As hipóteses asseguram, através da Proposição 19 que (4.15) define faceta para $P(S_H)$. Então, como $\chi(G) = \chi(G[V(S_H)]) \leq \chi(G[V(S_H \cup \{S\}]]) \leq \chi(G)$, para todo $S \in \mathcal{S} \setminus S_H$, a Proposição 14 garante o resultado desejado.*

□

O Corolário 6 mostra que subgrafos críticos máximos de G definem facetas de $P(\mathcal{S})$ através de (4.15). Assim, seria um passo natural a investigação dessas desigualdades para

subgrafos críticos quaisquer ou mesmo subgrafos críticos maximais, quer dizer, aqueles que não estão propriamente contidos em outros subgrafos críticos. Os exemplos das figuras 4.2 e 4.3 mostram que as desigualdades (4.15) relativas a esse tipo de subgrafos podem não definir facetas.

Figura 4.2: Exemplo de subgrafo crítico não-maximal que não induz faceta.

No caso grafo G da figura 4.2 é k -crítico com \bar{G} conexo. Logo, pela Proposição 18, $\sum_{S \in \mathcal{S}} \lambda_S \geq \chi(G) = 4$ define uma faceta de $P(\mathcal{S})$. Por outro lado, considerando $H = \{6, 7, 8, 9, 10\}$, temos que $G[H]$ é um buraco, ou seja, um subgrafo crítico, $\chi(G[H]) = 3$ e $\mathcal{S}_H = \mathcal{S} \setminus S$, onde $S = \{1, 2, 3, 4, 5\}$. Então, a desigualdade anterior somada a $\lambda_S \geq 1$ resulta na desigualdade (4.15) definida por H , mostrando que esta última não define faceta. Destacamos que $G[H]$ é subgrafo crítico não-maximal de G .

Já o grafo G apresentado figura 4.3, embora tenha o complemento conexo, não é k -crítico. Entretanto, para $H = \{1, 2, 3, 4, 5, 6, 7\}$ e $K = \{1, 2, 8, 9\}$, o buraco $G[H]$ e a clique $G[K]$ são subgrafos 3-crítico e 4-crítico maximais, respectivamente. Pela Proposição 19 e observando que $\chi(G) = 4$, concluímos que a desigualdade (4.15) relativa a K não define faceta de $P(\mathcal{S})$, uma vez que $\bar{G}[K]$ é desconexo. Adicionalmente, posto que $\mathcal{S}_H = \mathcal{S}$ e $\chi(G[H]) = \chi(G) - 1$, a desigualdade (4.15) associada a H não define faceta de $P(\mathcal{S})$ pois é claramente dominada por (4.15) referente a V .

Figura 4.3: Exemplo de subgrafo crítico maximal que não induz faceta.

4.3 Cálculo do Limite Inferior

Para calcularmos um limite inferior $\underline{\chi}(G)$ para o número cromático de G vamos usar a relaxação linear da formulação (CIM) acrescida de desigualdades do tipo (4.15). Precisamente, vamos considerar uma coleção \mathcal{H} de subconjuntos de V e determinar:

$$\underline{\chi}(G) = \min \sum_{S \in \mathcal{S}} \lambda_S \quad (4.23a)$$

$$\text{s.a: } \sum_{S \in \mathcal{S}_H} \lambda_S \geq \chi(G[H]), \quad \forall H \in \mathcal{H}, \quad (4.23b)$$

$$\lambda_S \geq 0, \quad \forall S \in \mathcal{S}. \quad (4.23c)$$

Sobre a composição do conjunto \mathcal{H} cabem alguns esclarecimentos. Primeiro, ele incluirá cada conjunto $\{v\}$, para $v \in V$, de modo que as restrições (4.9b), que definem facetas, segundo a Proposição 17, estejam todas presentes em (4.26). Segundo, \mathcal{H} compreenderá subconjuntos de vértices induzindo buracos e anti-buracos de G . Além disso, outros subgrafos críticos de G serão pesquisados, considerando repetidas operações de Mycielski a partir de buracos, anti-buracos e cliques de G , e seus conjuntos de vértices também comporão elementos de \mathcal{H} . Essas estruturas, como vimos na seção 4.2, podem também definir facetas de $P(\mathcal{S})$. Algoritmos para encontrar tais estruturas serão apresentados no próximo capítulo.

Como todas as desigualdades incluídas em (4.23b) são válidas para $P(\mathcal{S})$, claramente

o valor ótimo da formulação (4.23) fornece um limite inferior para o número cromático, conforme estabelece a seguinte proposição.

Proposição 20. $\chi(G) \geq \lceil \underline{\chi}(G) \rceil$.

Uma vez que a formulação (4.23) inclui todas as restrições (4.9b) e adicionalmente considera restrições sobre o número cromático de subestruturas de G , podemos dizer que $\underline{\chi}(G)$ combina os limites inferiores associados ao número cromático fracionário (Corolário 2) e ao número mínimo de cores necessárias para colorir subgrafos de G (Proposição 2), conferindo, assim, a seguinte qualidade ao novo limite inferior obtido.

Proposição 21. $\underline{\chi}(G) \geq \max\{\chi_F(G), \max_{H \in \mathcal{H}} \chi(G[H])\}$.

Para resolver (4.23) usaremos uma estratégia de geração simultânea de colunas e linhas (ver seção 3.3). Os cortes serão gerados pesquisando estruturas compreendidas pelo conjunto \mathcal{H} (detalhes serão apresentados na seção 5.2). Agora, concentremos a atenção no processo de geração colunas.

Consideremos um subconjunto $\mathcal{S}' \subseteq \mathcal{S}$ e o problema (4.23) restrito às colunas indexadas por \mathcal{S}' . Vamos denotar esse problema restrito por $R_{\mathcal{S}'}$ e seu valor ótimo por $\underline{\chi}(G, \mathcal{S}')$. Dado $\pi \geq 0$, o vetor de valores duais ótimos para $R_{\mathcal{S}'}$ associado a (4.23b), o custo reduzido de uma variável λ_S , $S \in \mathcal{S}$, é

$$\delta(S) = 1 - \pi(S),$$

onde

$$\pi(S) = \sum_{H: S \cap H \neq \emptyset} \pi_H \tag{4.24}$$

é peso do conjunto independente S com relação à solução dual π . Assim, o menor custo reduzido é

$$\Delta_{\mathcal{S}'} = 1 - \Pi_{\mathcal{S}'} \leq 0, \tag{4.25}$$

onde $\Pi_{\mathcal{S}'}$ é o peso do conjunto independente S^* que é solução do seguinte problema de conjunto independente ponderado máximo:

$$\Pi_{\mathcal{S}'} = \pi(S^*) = \max\{\pi(S) : S \in \mathcal{S}\}. \tag{4.26}$$

A nova coluna a ser gerada é definida por S^* , que será, portanto, incluído em \mathcal{S}' . A determinação de \mathcal{S}^* , como mostraremos no capítulo 5, pode ser feita através de uma

adaptação do algoritmo usado para resolver o problema original de conjunto independente máximo ponderado (4.11).

Como a determinação de $\underline{\chi}(G)$ é computacionalmente dispendiosa, dada a dificuldade de solução de (4.26), estudamos a seguir formas de abreviar o término do processo de geração de colunas. Especificamente, desejamos encontrar limites inferiores para $\chi(G)$ antes de encontrar a solução exata de (4.23), usando os seguintes resultados.

Proposição 22. *Seja $\mathcal{S}' \subseteq \mathcal{S}$. Se $\underline{\chi}(G) \leq U$ então $\underline{\chi}(G) \geq \underline{\chi}(G, \mathcal{S}') + U * \Delta_{\mathcal{S}'}$. Em particular, $(1 + \Delta_{\mathcal{S}'})\underline{\chi}(G, \mathcal{S}') \leq \underline{\chi}(G) \leq \underline{\chi}(G, \mathcal{S}')$.*

Prova: Seja $\mathcal{S}' \subseteq \mathcal{S}$. Denotemos por $RR_{\mathcal{S}}$ o problema obtido de $R_{\mathcal{S}}$ com a inclusão da restrição redundante $\sum_{S \in \mathcal{S}} \lambda_S \leq U$. Uma vez que $\underline{\chi}(G) \leq U$, $R_{\mathcal{S}}$ e $RR_{\mathcal{S}}$ têm o mesmo valor ótimo $\underline{\chi}(G)$. Consideremos também o dual de $RR_{\mathcal{S}}$, cujas restrições são

$$\sum_{H: S \cap H \neq \emptyset} \pi_H + \nu \leq 1, \quad \forall S \in \mathcal{S} \quad \pi \geq 0, \quad \nu \leq 0.$$

Por (4.24)–(4.26), temos que o vetor $\pi \geq 0$, solução ótima do dual de $R_{\mathcal{S}'}$, juntamente com $\nu = \Delta_{\mathcal{S}'}$ satisfazem as restrições acima, definindo, então, uma solução dual viável para $RR_{\mathcal{S}}$. Considerando o valor dessa solução dual e usando os teoremas de dualidade linear, segue-se que

$$\underline{\chi}(G) \geq \sum_H \pi_H + U * \Delta_{\mathcal{S}'} = \underline{\chi}(G, \mathcal{S}') + U * \Delta_{\mathcal{S}'}$$

Em particular, podemos tomar $U = \underline{\chi}(G, \mathcal{S}')$, pois $R_{\mathcal{S}'}$ é um problema restrito de $R_{\mathcal{S}}$. Assim, obtemos a segunda parte. \square

Corolário 7. *Seja $\mathcal{S}' \subseteq \mathcal{S}$. Se $\underline{\chi}(G) \leq U$ então $\lceil \underline{\chi}(G) \rceil \geq \lceil \underline{\chi}(G, \mathcal{S}') + U * \Delta_{\mathcal{S}'} \rceil$. Em particular, $\lceil (1 + \Delta_{\mathcal{S}'})\underline{\chi}(G, \mathcal{S}') \rceil \leq \lceil \underline{\chi}(G) \rceil \leq \lceil \underline{\chi}(G, \mathcal{S}') \rceil$.*

Assim, a partir da Proposição 20 e do Corolário 7, podemos obter um limite inferior para o número cromático a cada iteração do processo de geração de colunas.

Proposição 23. *Para todo $\mathcal{S}' \subseteq \mathcal{S}$, $\chi(G) \geq \lceil (1 + \Delta_{\mathcal{S}'})\underline{\chi}(G, \mathcal{S}') \rceil$.*

O Corolário 7 permite ainda a constatação de que o limite $\lceil \underline{\chi}(G) \rceil$ foi obtido antes mesmo de que a formulação (4.23) tenha sido resolvida exatamente, possibilitando também antecipar o processo de geração de colunas.

Proposição 24. *Seja $\mathcal{S}' \subset \mathcal{S}$ e defina $\tau_{\mathcal{S}'} = \lfloor \underline{\chi}(G, \mathcal{S}') \rfloor / \underline{\chi}(G, \mathcal{S}')$. Se $\Delta_{\mathcal{S}'} > \tau_{\mathcal{S}'} - 1$, ou equivalentemente, $\Pi_{\mathcal{S}'} < 2 - \tau_{\mathcal{S}'}$, então $\lceil \underline{\chi}(G) \rceil = \lceil \underline{\chi}(G, \mathcal{S}') \rceil$.*

Prova: Assuma que $1 + \Delta_{\mathcal{S}'} > \tau_{\mathcal{S}'}$. Então, $\lceil (1 + \Delta_{\mathcal{S}'}) \underline{\chi}(G, \mathcal{S}') \rceil \geq (1 + \Delta_{\mathcal{S}'}) \underline{\chi}(G, \mathcal{S}') > \lfloor \underline{\chi}(G, \mathcal{S}') \rfloor$. Logo, $\lceil (1 + \Delta_{\mathcal{S}'}) \underline{\chi}(G, \mathcal{S}') \rceil \geq \lceil \underline{\chi}(G, \mathcal{S}') \rceil$ e a igualdade segue pelo Corolário 7. \square

Duas observações merecem ser feitas com respeito aos limites inferiores estabelecidos na Proposição 23. Primeiro, eles não apresentam um comportamento monótono ao longo do processo de geração de colunas, quer dizer, o valor fornecido pela iteração seguinte pode ser maior ou menor que aquele gerado pela iteração corrente. Sendo assim, é conveniente manter o maior dos limites encontrados até o momento. Segundo, para a determinação de cada limite, faz-se necessário resolver exatamente o problema (4.26). Como o método de solução de (4.26) é implementado através de um processo enumerativo computacionalmente caro, opta-se normalmente por interromper tal processo quando for encontrado um conjunto independente que forneça uma coluna com custo reduzido “suficientemente” pequeno [29].

Nesse contexto, poderíamos ainda manter o processo enumerativo enquanto houver a possibilidade de um melhor limite inferior ser encontrado através da Proposição 23. Um critério de parada com esse propósito é apresentado pela proposição seguir, cuja demonstração é trivial.

Proposição 25. *Sejam $\mathcal{S}' \subset \mathcal{S}$ e $L \geq \mathbb{N}$ o melhor limite inferior conhecido para $\chi(G)$. Se $\Delta_{\mathcal{S}'} \leq L / \underline{\chi}(G, \mathcal{S}') - 1$, ou equivalentemente, $\Pi_{\mathcal{S}'} \geq 2 - L / \underline{\chi}(G, \mathcal{S}')$, então $\lceil (1 + \Delta_{\mathcal{S}'}) \underline{\chi}(G, \mathcal{S}') \rceil \leq L$.*

Capítulo 5

Algoritmos

5.1 Algoritmos Preliminares

A fim de resolver o problema de coloração de vértices, fazemos, primeiramente, um pré-processamento no grafo para obter algumas informações que nos guiarão nos processos seguintes, de indentificação de subgrafos críticos e de geração de colunas. Os algoritmos preliminares são realizados na seguinte ordem: de início, encontramos uma clique maximal qualquer, o que nos dá o primeiro limite inferior para o problema. A seguir fazemos um pré-processamento do grafo, retirando vértices trivialmente coloríveis. Então encontramos uma nova clique maximal no grafo reduzido e uma coloração viável para esse grafo usando o algoritmo *tabu search*: Então, encontramos todas as cliques maximais de tamanho igual àquele encontrado pela clique maximal inicial, para que possamos identificar, em seguida, operações de Mycielski sobre essas cliques.

5.1.1 Algoritmo para Clique Maximal

Em nossa implementação, precisamos inicialmente que algumas cliques maximais seja encontradas para que possamos encontrar estruturas de Mycielski a partir delas, ou seja, dada uma clique $K \subseteq G$, procuramos $M(K) \subseteq G$, conforme visto no Capítulo 2. Quanto maior a clique, melhor, pois esta fornecerá, caso seja encontrado uma estrutura de Mycielski, um limite inferior inicial ainda melhor para o problema. Este limite inferior também será usado no pré-processamento do grafo. Inicialmente encontramos uma clique maximal qualquer, utilizando uma heurística simples para clique maximal. Essa heurística consiste

em, inicialmente, ordenar os vértices segundo seu grau. Então inclui-se na clique maximal K o primeiro vértice v da ordem e remove-se da ordem os vértices não adjacentes a esse vértice, incluindo o próprio v . Repete-se o processo até que não haja mais vértices na ordem. A sinopse deste algoritmo está descrita nos algoritmos 1 e 2.

Algoritmo 1 *CliqueMaximal*(G)

Entrada: Grafo $G = (V, E)$

Saída: Clique maximal K

- 1: $K \leftarrow \emptyset$
 - 2: $S \leftarrow V(G)$
 - 3: Ordenar S pelo grau dos vértices.
 - 4: **Retorna** *CliqueMaximalRecursiva*(G, K, S)
-

Algoritmo 2 *CliqueMaximalRecursiva*(G, K, S)

Entrada: Grafo G , Clique K , Conjunto ordenado S

Saída: Clique maximal K com respeito ao subgrafo $G[S]$

- 1: $v \leftarrow S[1]$
 - 2: $S \leftarrow S \cap N(v)$
 - 3: $K \leftarrow K \cup \{v\}$
 - 4: **se** $S \neq \emptyset$ **então**
 - 5: **Retorna** *CliqueMaximalRecursiva*(G, K, S)
 - 6: **senão**
 - 7: **Retorna** K
 - 8: **fim se**
-

5.1.2 Algoritmo para todas as cliques maximais de tamanho fixo

A partir da clique determinada nos Algoritmo 1 e 2, o nosso objetivo agora é encontrar um conjunto de subgrafos que podem gerar um limite inferior para o problema. Para isso, iremos procurar cliques maximais com tamanho igual ao tamanho dessa clique. A partir dessas cliques maximais, procuraremos um subgrafo de Mycielski que contém cada uma delas.

Utilizaremos o algoritmo descrito por Byskov [3] para conjuntos independentes maximais de tamanho fixo c . Como um conjunto independente é uma clique no grafo comple-

mento, fizemos uma adaptação desse algoritmo, encontrando assim cliques maximais do grafo. Os algoritmos 3 e 4 trazem o pseudo-código da descrição abaixo.

Uma breve descrição do algoritmo: iniciamos o algoritmo com os conjuntos $K = V(G)$ e um valor c para o qual queremos encontrar as cliques maximais. A cada passo checamos se o conjunto K é uma clique de tamanho c . Caso seja, guardamos esta clique, caso não seja e $|K| > c$, fazemos duas chamadas recursivas do algoritmo: chamamos o algoritmo incluindo um vértice v em K e portanto o restante do grafo a ser considerado é a vizinhança de v , assim fazendo $G' = K \cap N[v]$; e não incluímos o vértice v e fazemos $G' = K \setminus \{v\}$.

Algoritmo 3 *AcharTodasCliquesMaximais*(G, c)

Entrada: Grafo $G = (V, E)$, Inteiro c

Saída: Conjunto de cliques maximais de tamanho c

- 1: $K \leftarrow V(G)$
 - 2: Ordenar K pelo grau dos vértices.
 - 3: *AcharTodasMaximaisRec*($G, K, c, 1$)
-

Algoritmo 4 *AcharTodasMaximaisRec*(G, K, c, i)

Entrada: Grafo G , Conjunto ordenado K , Inteiro c , Índice i

Saída: Cliques maximais em $G[K]$ de tamanho c

- 1: **se** $|K| = c$ **então**
 - 2: *checar*(K)
 - 3: **senão**
 - 4: **se** $|K| > c$ **então**
 - 5: $v \leftarrow K[i]$
 - 6: *AcharTodasMaximaisRec*($G, K \cap N[v], c, i + 1$)
 - 7: *AcharTodasMaximaisRec*($G, K \setminus \{v\}, c, i$)
 - 8: **fim se**
 - 9: **fim se**
-

Na prática, para grafos densos, o número de cliques enumeradas por esse algoritmo pode ser bastante grande, por isso paramos o algoritmo quando encontramos um número fixo de cliques. Escolhemos, por experiência, parar quando encontramos m cliques, onde m é o número de arestas do grafo.

5.1.3 Algoritmo para pré-processamento

Podemos simplificar um grafo para o problema de coloração antes de aplicarmos qualquer método para a resolução do mesmo. Isso é bastante útil para reduzir o tamanho do problema e tornar métodos complexos aplicáveis na prática com os problemas reduzidos. No caso da coloração de grafos, podemos identificar certos vértices que podem ser removidos para simplificarmos o processamento. Eles se dividem em 3 categorias: vértices universais, vértices dominados e vértices de grau baixo.

Um vértice universal é um vértice adjacente a qualquer outro vértice do grafo. Os vértices universais do grafo podem ser removidos pois certamente vão precisar de uma cor adicional exclusiva. Portanto, retiramos tais vértices e após a resolução do problema restrito ao grafo menor, acrescentamos uma cor adicional para cada vértice universal removido.

Um vértice dominado u é aquele cuja vizinhança está contida na vizinhança de um outro vértice v que não é adjacente a ele. Daí, claramente, o vértice dominado u pode receber, sem qualquer prejuízo, em qualquer solução ótima a cor do vértice v . Os vértices dominados são retirados do grafo e repostos após a solução do problema restrito ao grafo sem os vértices retirados no pré-processamento sem aumentar o número cromático do grafo original.

Por fim, dizemos que um vértice tem grau baixo quando o seu grau é menor que um limite inferior conhecido para o problema. Observe que os vizinhos de tal vértice vão receber no máximo um número de cores menor do que o limite inferior, o que faz sobrar uma cor para o vértice de grau baixo. Observe que a retirada do vértice de grau baixo não compromete o limite inferior do problema. Portanto a cada retirada do vértice de grau baixo permanecemos com o mesmo limite inferior para o problema.

Observe ainda que, ao retirarmos esses vértices, podemos tornar outros vértices, que a princípio não se encaixavam em nenhuma dessas classes, em vértices universais, dominados ou de grau baixo. Portanto o procedimento de retirada de vértices é repetido até que não haja mais qualquer vértice em uma das classes (universais, dominados ou de grau baixo). Essas técnicas de pré-processamento foram apresentadas em [27].

O próximo passo é verificar se G' é conexo. A retirada dos vértices pode ter desconectado o grafo original. O problema agora é determinar o número cromático das componentes conexas de G' . Lembre-se que o número cromático da maior das componentes é o

Algoritmo 5 *PreProcessamento*(G, l)

Entrada: Grafo G , Limite inferior l **Saída:** Grafo processado G' , Limite inferior l para G'

- 1: **Repita**
 - 2: Remova vértices universais em G
 - 3: Remova vértices dominados em G
 - 4: Remova vértices de grau baixo em G
 - 5: Atualize o limite inferior l
 - 6: **Até que** Nenhum vértice tenha sido removido nesta iteração
 - 7: **Retorna** G
-

número cromático de G' .

5.2 Algoritmos para Geração de Cortes

Para a geração de cortes, concentramos nossa atenção em cortes advindos de estruturas como buracos e anti-buracos ímpares e construções a partir de buracos e anti-buracos ímpares. Inicialmente, utilizamos uma heurística para encontrar buracos ímpares e em seguida fizemos uma extensão dessa heurística para que pudéssemos encontrar subgrafos gerados por operações de Mycielsky, conforme visto na seção 2.4.1 . A razão para a busca de tais estruturas reside na idéia do teorema forte dos grafos perfeitos, ou seja, como as restrições de clique estão satisfeitas pela definição da formulação, então nos deteremos a tais estruturas.

Vejamos agora o algoritmo de geração de buracos e anti-buracos. Hoffman e Padberg [23] desenvolveram uma heurística para identificação de buracos ímpares enquanto estudavam o problema de escalonamento de tripulação em companhias aéreas. A heurística consiste no seguinte: inicialmente, construa um grafo de níveis enraizado em um dado vértice v . Cada outro vértice u do grafo tem um nível igual a distância, em arestas, entre u e v . Agora, tomemos cada aresta (u, w) onde os vértices u e w estão no mesmo nível. Sabemos que u e w estão à mesma distância k de v e que possuem uma areta entre eles. Para verificar se o caminho de u até v mais o caminho de v até w mais a areta (u, w) é um buraco, temos que nos certificar que não há cordas entre os caminhos. Portanto o próximo passo do algoritmo é remover os vizinhos dos vértices do caminho de u até v

(removendo o próprio caminho) do grafo. Em seguida, procuramos um caminho entre v e w no grafo restante, passando somente por vértices que estejam num nível inferior ou igual ao nível original de w . Caso exista, podemos guardar esse buraco e partir para a próxima aresta.

Observe que este método pode ser aplicado a cada vértice, mas que esse processo gera muitos buracos redundantes, ou seja, encontramos o mesmo buraco possivelmente várias vezes e portanto é preciso limpar a lista de buracos gerados no final do processo. Para gerar anti-buracos, aplicamos o algoritmo ao grafo complementar \bar{G} .

Algoritmo 6 *EncontrarBuracos*($G, nivel_{min}, nivel_{max}$)

Entrada: Grafo $G = (V, E)$, Nível mínimo $nivel_{min}$, Nível máximo $nivel_{max}$

Saída: Coleção \mathfrak{S} de buracos em G de tamanho entre $2(nivel_{min}) + 1$ a $2(nivel_{max} + 1)$.

```

1: para todo  $v \in V(G)$  faça
2:   CaminhoMinimo( $G, v, dist1, caminho1$ )
3:   para  $k = nivel_{min} .. nivel_{max}$  faça
4:     para todo  $(w, z) \in E(G)$  tal que  $w$  e  $z$  estão no nível  $k$  faça
5:        $G' = G \setminus \{u : u \in caminho1(w, v)\}$ 
6:        $G' = G' \setminus \{u : dist1[u] > k\}$ 
7:       CaminhoMinimo( $G', v, dist2, caminho2$ )
8:       se  $caminho2(z, v) \neq \emptyset$  então
9:          $\mathfrak{S} \leftarrow \mathfrak{S} \cup Buraco(caminho1(w, v), caminho2(z, v), (w, z))$ 
10:      fim se
11:    fim para
12:  fim para
13: fim para
14: Retorna  $\mathfrak{S}$ 

```

Para o nosso trabalho, fizemos ainda uma extensão desse algoritmo baseada na seguinte observação: um $C5$ é um grafo de Mycielsky que pode ser encontrado pela heurística anterior quando se procura arestas no segundo nível. Ora, mas ao invés de uma aresta, podemos procurar outras estruturas no segundo nível. A princípio, podemos procurar qualquer subgrafo H . A mudança ocorre no seguinte ponto: ao invés de acharmos um caminho dos vértices até o nó raiz z , precisamos encontrar um conjunto independente U no nível 1 de z que satisfaça a condição seguinte:

Para todo $v \in H$, existe $u \in U$ tal que $N_H(u) = N_H(v)$.

onde $N_H(v)$ representa a vizinhança do vértice v no subgrafo H .

Algoritmo 7 IdentificarMycielsky(G, H, z)

Entrada: Grafo G , Subgrafo H a ser identificado, Raiz z

Saída: Grafo identificado U caso exista, ou \emptyset caso contrário

- 1: Construa o grafo de níveis de G com raiz em z
- 2: Sejam N_1 e N_2 os níveis 1 e 2 desse grafo.
- 3: $U \leftarrow \emptyset$
- 4: **se** $H \subseteq N_2$ **então**
- 5: **para todo** $v \in H$ **faça**
- 6: Encontrar $u \in N_1$ tal que $N_H(u) = N_H(v)$ e $N_U[u] = \emptyset$
- 7: **se** existe u **então**
- 8: $U \leftarrow U \cup \{u\}$
- 9: **senão**
- 10: **Retorna** \emptyset
- 11: **fim se**
- 12: **fim para**
- 13: **fim se**
- 14: **Retorna** U

Em nossa implementação, utilizamos o algoritmo acima da seguinte forma: iniciamos com uma base de cliques maximais, buracos e anti-buracos encontrados pelos métodos anteriores. Aplicamos o método acima, buscando essas estruturas (possivelmente arestas) no nível 2 de algum vértice. Então geramos um novo conjunto de estruturas que irá retro-alimentar o algoritmo para procurarmos estruturas cada vez maiores.

5.3 Algoritmos para Geração de Colunas

Conforme citado em [29], Mehrotra e Trick utilizam uma versão ponderada do algoritmo de Pardalos e Carragham para encontrar o conjunto independente máximo ponderado a cada iteração da geração de colunas. Esse algoritmo consiste no seguinte. Primeiro é dada uma ordenação aos vértices de acordo com o "peso" de sua anti-vizinhança. Em

Algoritmo 8 *EncontrarMycielski(G, \mathfrak{S})*

Entrada: Grafo G , Coleção de Subgrafos \mathfrak{S} **Saída:** Nova coleção de subgrafos \mathfrak{S}' de G gerados por operações de Mycielski em \mathfrak{S}

- 1: **para todo** $v \in V(G)$ **faça**
 - 2: **para todo** $H \in \mathfrak{S}$ **faça**
 - 3: $U \leftarrow \text{IdentificarMycielski}(G, H, v)$
 - 4: **se** $U \neq \emptyset$ **então**
 - 5: $\text{GuardarSubgrafo}(H, U, v)$
 - 6: **fim se**
 - 7: **fim para**
 - 8: **fim para**
 - 9: **Retorna** \mathfrak{S}'
-

Algoritmo 9 *EncontrarMycielskiIterativo(G, \mathfrak{S})*

Entrada: Grafo G , Coleção de Subgrafos \mathfrak{S} **Saída:** Nova coleção de subgrafos \mathfrak{S}' de G gerados por operações de Mycielski iterativas

- 1: **Repita**
 - 2: $\mathfrak{S} \leftarrow \text{EncontrarMycielski}(G, \mathfrak{S})$
 - 3: $\mathfrak{S}' = \mathfrak{S}' \cup \mathfrak{S}$
 - 4: **Até que** $\mathfrak{S} = \emptyset$
 - 5: **Retorna** \mathfrak{S}'
-

seguida iniciamos um conjunto $S = \emptyset$. A partir daí tomamos dois caminhos: ou incluímos um vértice em S e trabalhamos com sua anti-vizinhança; ou não incluímos o vértice e continuamos com o mesmo grafo exceto pelo vértice recém analisado. Essa árvore de decisão é uma enumeração de todos os conjuntos independentes possíveis. Para evitar conjuntos independentes redundantes, ou seja, conjuntos independentes iguais a menos da ordem em que os vértices são considerados, percorremos os vértices segundo a ordem pré-estabelecida e em profundidade (para garantir uma solução corrente o quanto antes), considerando somente os vértices que não tenham qualquer vizinho em S . Antes de incluirmos um vértice v é possível fazer uma estimativa de quanto podemos ganhar por incluí-lo no conjunto S . Caso o peso de S mais o peso de $N^+[v]$ seja menor que a melhor solução conhecida, então podemos descartar a inclusão desse vértice, onde $N^+[v]$ são os vértices da anti-vizinhança de v que vêm depois dele na ordem. Com essa regra de poda,

é possível reduzir a quantidade de nós enumerados no processo.

Algoritmo 10 $MIS(G, w)$

Entrada: Grafo $G = (V, E)$, pesos w , peso mínimo w_{min} e peso suficiente w_{suf}

Saída: Conjunto Independente S

- 1: $S \leftarrow \emptyset$
 - 2: $I \leftarrow V(G)$
 - 3: Ordenar I pelo $\sum_{u \in \bar{N}[v]} w(u)$ para cada $v \in I$
 - 4: **para todo** $v \in I$ **faça**
 - 5: $S \leftarrow MISRecursivo(G, I, S, w, w_{min}, w_{suf})$
 - 6: **fim para**
 - 7: **Retorna** S
-

Embora seja um processo enumerativo, o Algoritmo 10 certamente resolve o conjunto independente máximo ponderado com uma vantagem: é possível encontrar um valor mínimo para o conjunto independente máximo ponderado de tal forma a cortar soluções que nunca alcançariam este valor e também é possível dar um valor suficiente tal que caso o algoritmo encontre um conjunto independente maior que aquele valor, já é suficiente para o processo de geração de colunas. É importante frisarmos que estes dois valores podem reduzir e muito o tempo que se gasta nesse processo de resolver o CIMP, que é, como pudemos comprovar experimentalmente, a parte crítica do processo de geração de colunas, a parte que toma mais tempo.

No entanto, o algoritmo de Pardalos e Carranham não resolve o problema visto em nossa formulação, quando temos restrições estruturais. Para isso, precisamos desenvolver uma extensão do algoritmo anteriormente citado de forma a contemplar esse novo aspecto do problema e conservar as medidas de valor mínimo e suficiente para o CIMP, conforme descrito acima.

Nosso problema agora trata-se de encontrar um conjunto independente que seja máximo em relação ao peso dos vértices e ao peso das estruturas. Devemos considerar, ao incluirmos um vértice em S quais estruturas ele vai tocar. O algoritmo inicia, portanto, fazendo uma ordenação dos vértices segundo seu peso mais o peso das estruturas que ele toca. A idéia é manter a mesma heurística de ordenação dos vértices, considerando agora estruturas maiores. E como no algoritmo anterior, vamos construir a árvore de decisão segundo essa ordem e percorrê-la em profundidade, garantindo uma solução viável

o quanto antes. Ao incluirmos um vértice u em S devemos somar ao peso de S não só o peso do vértice u , mas o peso das estruturas que ele toca que não foram tocadas por qualquer outro vértice $v \in S$. A estimativa para a poda de ramos é feita da seguinte forma: caso o peso de S mais o peso de $N^+[u]$, onde u é o vértice em questão, mais o peso das estruturas em que u faz parte, mas que nenhum outro vértice $v \in S$ faz parte seja menor que a melhor que o peso da melhor solução corrente, pode-se desconsiderar o vértice u .

O Algoritmo 12 resolve o problema do conjunto independente máximo com estruturas de bonificação e por conservar a mesma pesquisa do algoritmo de Carraghan e Pardalos [7], ele consegue manter a idéia dos valores mínimo e limite, conforme vistos acima.

5.4 Algoritmo Geral

Nesta seção vamos resumir todo o processo de geração de cortes e colunas em um algoritmo para melhor entendimento. Seu pseudo-código 14 está escrito em passos largos, que merecem uma melhor explicação.

A primeira linha em nossa implementação refere-se a encontrar uma clique maximal com o Algoritmo 1. Então fazemos na segunda linha um pré-processamento do grafo com o Algoritmo 5. A seguir um algoritmo "tabu-search"[22] é usado para fornecer uma coloração viável e dar um limite superior para o problema. A partir daí, é encontrado um novo limite inferior através de uma clique maximal e então os subgrafos críticos são procurados no grafo através dos algoritmos vistos na Seção 5.2, de acordo com a linha 4. Na linha 5, construímos a coloração inicial e resolvemos o modelo pela primeira vez. Em nossa implementação, caso o limite inferior obtido seja igual ao limite superior, nós paramos a execução nesse ponto pois a solução ótima foi alcançada. A partir da linha 6 começa o processo de geração de linhas e colunas para problema. As técnicas para resolver este problema foram vistas na Seção 3.3. Além disso, aplicamos a Propriedade 20 para atualizar o limite inferior e possivelmente abreviar o processo de geração de colunas.

Algoritmo 11 $MISRecursivo(G, I, S, w, w_{min}, w_{suf})$

Entrada: Grafo G , Conjunto I de vértices a serem considerados, Conjunto Independente parcial S , pesos dos vértices w , peso mínimo w_{min} e peso suficiente w_{suf}

Saída: Conjunto Independente S

```
1:  $w^* \leftarrow \sum_{u \in S} w(u)$ 
2: se  $w^* \geq w_{suf}$  então
3:   Retorna  $S$ 
4: fim se
5: se  $I = \emptyset$  então
6:   se  $w^* \geq w_{min}$  então
7:      $w_{min} \leftarrow w^*$ 
8:     Retorna  $S$ 
9:   fim se
10: senão
11:    $v \leftarrow I[1]$ 
12:    $S \leftarrow S \cup \{v\}$ 
13:    $I \leftarrow I \cap \bar{N}(v)$ 
14:   para todo  $z \in I$  faça
15:     se  $w^* + \sum_{u \in \bar{N}(z)} w(u) > w_{min}$  então
16:        $S \leftarrow MISRecursivo(G, I, S, w, w_{min}, w_{suf})$ 
17:     fim se
18:   fim para
19: fim se
```

Algoritmo 12 $MISAdaptado(G, \mathfrak{S}, w, W, w_{min}, w_{suf})$

Entrada: Grafo G , Subgrafos \mathfrak{S} , pesos dos vértices w , pesos dos subgrafos W , peso mínimo w_{min} e peso suficiente w_{suf}

Saída: Conjunto Independente S

- 1: $S \leftarrow \emptyset$
 - 2: $T \leftarrow \emptyset$
 - 3: $I \leftarrow V(G)$
 - 4: Ordenar I pelo $\sum_{u \in \bar{N}[v]} w(u) + \sum_{H \in \mathfrak{S}: v \in H} W(H)$ para cada $v \in I$
 - 5: **para todo** $v \in I$ **faça**
 - 6: $S \leftarrow MISAdaptadoRecursivo(G, \mathfrak{S}, I, S, T, w, W, w_{min}, w_{suf})$
 - 7: **fim para**
 - 8: **Retorna** S
-

Algoritmo 13 *MISAdaptadoRecursivo*($G, \mathfrak{S}, I, S, T, w, W, w_{min}, w_{suf}$)

Entrada: Grafo G , Subgrafos \mathfrak{S} , Conjunto I de vértices a serem considerados, ConjuntoIndependente parcial S , Conjunto de estruturas T tocadas pelos vértices de S , pesos dos vértices w , pesos dos subgrafos W , peso mínimo w_{min} e peso suficiente w_{suf} **Saída:** Conjunto Independente S

- 1: $w^* \leftarrow \sum_{u \in S} w(u) + \sum_{H \in \mathfrak{S} \setminus T: v \in H} W(H)$
 - 2: **se** $w^* \geq w_{suf}$ **então**
 - 3: **Retorna** S
 - 4: **fim se**
 - 5: **se** $I = \emptyset$ **então**
 - 6: **se** $w^* \geq w_{min}$ **então**
 - 7: $w_{min} \leftarrow w^*$
 - 8: **Retorna** S
 - 9: **fim se**
 - 10: **senão**
 - 11: $v \leftarrow I[1]$
 - 12: $S \leftarrow S \cup \{v\}$
 - 13: $T \leftarrow T \cup \{H \in \mathfrak{S} : v \in H\}$
 - 14: $I \leftarrow I \cap \bar{N}(v)$
 - 15: **para todo** $z \in I$ **faça**
 - 16: **se** $w^* + \sum_{u \in \bar{N}(z)} w(u) + \sum_{H \in \mathfrak{S} \setminus T: z \in H} W(H) > w_{min}$ **então**
 - 17: $S \leftarrow MISRecursivo(G, \mathfrak{S}, I, S, T, w, W, w_{min}, w_{suf})$
 - 18: **fim se**
 - 19: **fim para**
 - 20: **fim se**
-

Algoritmo 14 *LimiteInferior*(G)

Entrada: Grafo G

Saída: Limite inferior l para o problema de coloração em G

- 1: Encontre um limite inferior para o grafo G
 - 2: Faça o pré-processamento do grafo G e para cada uma de suas componentes conexas execute os passos seguintes.
 - 3: **para todo** $T_i(G) \subseteq G$ componente conexa de G **faça**
 - 4: Encontre um limite superior LS_i e um limite inferior LI para a componente $T_i(G)$
 - 5: Procure todos os subgrafos de Mycielski a partir de uma base de cliques e buracos e atualize o limite inferior LI .
 - 6: Faça a formulação a partir de uma coloração inicial e resolva, guardando seu valor em z^*
 - 7: **Repita**
 - 8: Verifique quais cortes estão violados baseados nas estruturas encontradas
 - 9: Encontre uma nova coluna resolvendo o conjunto independente máximo
 - 10: Atualize o limite inferior LS
 - 11: Resolva a formulação e armazene o valor em z^* .
 - 12: **Até que** $LI == \lceil z^* \rceil$
 - 13: **fim para**
 - 14: $LI^* \leftarrow \max\{LI_i : T_i(G) \text{ componente conexa de } G\}$
 - 15: **Retorna** LS^*
-

Capítulo 6

Resultados

Os algoritmos apresentados no capítulo anterior foram implementados e testados em uma série de instâncias extraídas de DIMACS [25] e serão descritas na seção a seguir. Em seguida faremos uma análise crítica dos resultados considerando as tabelas apresentadas.

6.1 Descrição das Instâncias

As instâncias escolhidas dentro do conjunto de instâncias de DIMACS foram aquelas em que o tamanho da maior clique é estritamente menor que o valor do melhor limite superior conhecido para o problema.

Grafos aleatórios (DSJC)

Grafos aleatórios gerados por David Johnson e usados em seu artigo com Aragon, McGeoch, and Schevon [24]. Cada grafo DSJC.n.p é criado a partir de um par (n, p) , onde n é o número de vértices do grafo e p é a probabilidade de existir uma aresta entre quaisquer dois vértices.

Grafos Rainha (QUEEN)

Dado um tabuleiro de xadrez com medidas $q \times r$, o grafo queen.q_r é um grafo com qr vértices, onde cada vértice está associado a uma célula do tabuleiro e uma aresta existe entre dois vértices se uma rainha no xadrez pode se deslocar entre as células correspondentes, ou seja, se eles estão na mesma linha, coluna ou diagonal. Resolver o problema de coloração nestes grafos é equivalente a responder se q conjuntos de r rainhas podem ser dispostos sobre um tabuleiro de xadrez sem que qualquer uma

delas seja visível a qualquer outra. A resposta é positiva se o número cromático do grafo é r .

Grafos de Mycielski (myciel, insertions,)

Grafos construídos a partir de operações de Mycielski (ver seção 2.4), estes são grafos construídos a partir de operações de Mycielski e possivelmente a adição de outras arestas. A classe `myciel.x` compreende os grafos de Mycielski conforme originalmente definidos [26]. As instâncias `FullIns` e `Insertions` são geradas a partir de construções generalizadas dos grafos de Mycielski.

Outros Grafos (FPSOL, MUG)

São outros grafos quem compoem as instância da biblioteca DIMACS e que estão nos padrões de tamanho para os quais nosso trabalho comporta a resolução.

6.2 Detalhes da Implementação

Os algoritmos vistos nessa dissertação foram implementados em um microcomputador Pentium IV 2.0GHz com 1GB de memória RAM, e a biblioteca CPLEX9.0 foi utilizada para resolver problemas de programação linear. Um limite de tempo fixo de duas horas foi estabelecido para cada instância.

O limite superior do grafo, bem como a coloração inicial para a geração de colunas foi dada pela heurística `tabu-search` descrita em [22].

A geração de colunas é o passo mais caro em termos de tempo computacional durante todo o processo. A cada iteração é preciso resolver o problema do conjunto independente máximo com algumas modificações. Essas modificações são tamanhos mínimos e suficientes para o problema do CIM. Esses limites permitem uma parada antecipada da geração de uma coluna, visto que podem podar nós da árvore de recursão dos algoritmos implementados para este problema. Ainda assim, os algoritmos recursivos tomavam bastante tempo e por isso precisamos criar critérios de parada alternativos para o processo de geração de colunas como um todo, conforme a propriedade 20. Outros autores ([29], [17]) já haviam verificado que, no final do processo, o custo de geração da coluna se torna alto e é pouco efetivo. Nesses casos, as modificações sugeridas na proposição 20 se tornam não só válidas, mas preciosas.

Caso um grafo tenha sido separado em componentes, as informações referentes a ele contidas na tabela se referem a componente com maior número cromático.

6.3 Tabelas de Resultados

Em nosso trabalho, nos concentramos em verificar a qualidade do limite inferior produzido pelos métodos apresentados nos capítulos anteriores. Para cada grafo, as seguintes informações foram analisadas e serão aqui mostradas.

Na tabela 1 temos os resultados referentes aos limites inferiores obtidos.

Tamanho - O número de vértices n e o número de arestas m .

Limites - Quatro limites são apresentados. LSI é o limite superior inicial, enquanto LII é o limite inferior inicial, dado pelo maior $\chi(H)$, $H \subseteq G$. Durante o processo de geração de colunas, atualizamos a cada iteração LSC e LIC que são os limites superior (dado pela resolução do problema) e inferior contínuos (calculado pela propriedade 20).

Resultado - Limite inferior inteiro ao final da execução, número cromático fracionário $\chi_F(G)$ e o número cromático $\chi(G)$ do grafo, caso seja conhecido.

Nome	Tamanho		Limites				Resultado		
	n	m	LSI	LII	LSC	LIC	LI	$\chi_F(G)$	$\chi(G)$
DSJC125.1	125	1472	6	4	5.23	3.87	4	4.86	5
DSJC125.5	125	7782	19	9	15.75	15.00	16	15.73	?
DSJC125.9	125	13922	50	27	42.73	42.00	43	42.73	?
DSJC250.1	250	6436	10	4	9.34	3.75	4	7.94	8
DSJC250.5	250	31336	32	9	25.16	25.00	26	?	?
DSJC250.9	250	55794	86	34	70.39	70.00	71	?	?
queen5.5	25	320	5	5	5.00	5.00	5	5.00	5
queen6.6	36	580	8	5	7.00	7.00	7	7.00	7
queen7.7	49	952	9	5	7.00	7.00	7	7.00	7
queen8.8	64	1456	10	5	8.44	8.00	9	8.44	9
queen8.12	96	2736	12	5	12.00	12.00	12	12.00	12
queen9.9	81	2112	11	5	9.00	9.00	9	9.00	10
queen10.10	100	580	12	5	10.00	9.4	10	10.00	10
queen11.11	121	580	13	5	11.00	11.00	11	11.00	11
FPSOL2.i.1	496	11654	65	55	65.00	65.00	65	65.00	65
FPSOL2.i.2	451	8691	30	29	30.00	30.00	30	30.00	30
FPSOL2.i.3	425	8688	30	29	30.00	30.00	30	30.00	30
MUG88.1	88	146	4	3	3.21	2.99	3	?	4
MUG88.25	88	146	4	3	3.19	2.99	3	?	4
MUG100.1	100	166	4	3	3.19	2.98	3	?	4
MUG100.25	100	166	4	3	3.21	2.99	3	?	4

Nome	Tamanho		Limites				Resultado		
	n	m	LSI	LII	LSC	LIC	LI	$\chi_F(G)$	$\chi(G)$
myciel3	11	20	4	4	4.00	4.00	4	2.90	4
myciel4	23	71	5	5	5.00	5.00	5	3.20	5
myciel5	47	236	6	6	6.00	6.00	6	3.55	6
myciel6	95	755	7	7	7.00	7.00	7	3.80	7
FullIns1.3	30	100	4	4	4.00	4.00	4	3.33	4
FullIns1.4	93	593	5	5	5.00	5.00	5	3.63	?
FullIns1.5	282	3247	6	6	6.00	6.00	6	4.02	?
FullIns2.3	52	201	5	5	5.00	5.00	5	4.25	5
FullIns2.4	212	1621	6	6	6.00	6.00	6	4.56	?
FullIns3.3	80	346	6	6	6.00	6.00	6	5.20	6
FullIns4.3	114	541	7	7	7.00	7.00	7	6.22	7
FullIns5.3	154	792	8	8	8.00	8.00	8	7.20	8
Insertions1.4	67	232	5	3	3.06	2.99	3	2.84	4
Insertions1.5	202	1227	6	3	4.42	2.98	3	3.32	?
Insertions2.3	37	72	4	3	3.00	3.00	3	?	4
Insertions2.4	149	541	5	3	3.39	2.92	3	2.79	4
Insertions3.3	56	110	4	3	3.00	3.00	3	?	4
Insertions3.4	281	1046	5	3	4.14	2.94	3	2.80	?
Insertions4.3	79	156	4	3	3.00	3.00	3	2.38	4

Tabela 1. Mostra a relação entre os limites.

A tabela 2 traz as informações relativas ao tempo gasto e ao crescimento do modelo.

Tamanho - O número de vértices n e o número de arestas m .

Tempos - Em nossa implementação, medimos três tempos. O tempo de pré-processamento do grafo T_{PP} , o tempo de encontrar os subgrafos que geram os cortes T_S , e o tempo gasto na geração de colunas T_{GC} . Todos os tempos estão em segundos.

Modelo - Ao final da execução, o modelos tem col colunas e lin linhas.

Nome	Tamanho		Tempos			Modelo	
	n	m	T_{PP}	T_S	T_{GC}	col	lin
DSJC_125.1	125	1472	8	20	7972	3585	125
DSJC_125.5	125	7782	7	3	40	822	125
DSJC_125.0	125	13922	4	35	1	319	125
DSJC_250.1	250	6436	30	206	7237	5473	250
DSJC_250.5	250	31336	32	121	4684	2017	250
DSJC_250.9	250	55794	15	34	11	786	250
queen5.5	25	320	1	0	0	84	26
queen6.6	36	580	2	1	0	129	36
queen7.7	49	952	2	1	1	220	49
queen8.8	64	1456	4	19	2	304	64
queen8.12	96	2736	5	158	0	12	96
queen9.9	81	2112	6	10	13	607	81
queen10.10	100	580	7	236	127	731	100
queen11.11	121	580	7	51	681	903	121
FPSOL2.i.1	496	11654	100	26	0	411	241
FPSOL2.i.2	451	8691	133	27	24	1370	300
FPSOL2.i.3	425	8688	107	27	55	2630	300
MUG88.1	88	146	6	0	7644	1132	110
MUG88.25	88	146	5	0	75	1060	107
MUG100.1	100	166	7	1	9310	3162	110
MUG100.25	100	166	6	0	9977	1004	114

Nome	Tamanho		Tempos			Modelo	
	n	m	T_{PP}	T_S	T_{GC}	col	lin
myciel3	11	20	0	0	0	4	12
myciel4	23	71	1	0	0	5	24
myciel5	47	236	2	5	0	6	48
myciel6	95	755	7	286	0	7	96
FullIns1.3	30	100	1	0	0	4	36
FullIns1.4	93	593	5	9	0	5	99
FullIns1.5	282	3247	48	4493	0	6	288
FullIns2.3	52	201	2	0	0	5	53
FullIns2.4	212	1621	16	309	0	6	213
FullIns3.3	80	346	6	7	0	6	79
FullIns4.3	114	541	12	4	0	7	113
FullIns5.3	154	792	16	25	0	8	152
Insertions1.4	67	232	3	3	7301	696	631
Insertions1.5	202	1227	24	2557	8385	1319	208
Insertions2.3	37	72	1	0	1	233	55
Insertions2.4	149	541	9	62	7687	1064	236
Insertions3.3	56	110	3	0	2	424	80
Insertions3.4	281	1046	65	905	7201	1618	282
Insertions4.3	79	156	6	0	12	689	94

Tabela 2. Relações entre tempo e tamanho dos grafos.

Em seguida, na tabela 3 é possível vermos as informações sobre os subgrafos encontrados em cada um dos grafos de teste.

Coloração - Três valores são apresentados. $\chi_C(G)$ é o número cromático da clique maximal inicialmente encontrada. $\chi_H(G)$ é o maior número cromático entre todos os subgrafos encontrados em G . E por último $\chi(G)$ é o número cromático de G , caso conhecido.

Subgrafos - Diversos parâmetros foram analisados com relação ao número de subgrafos.

O primeiro SA é o número de subgrafos que foram encontrados cujos cortes foram

adicionados ao modelo. SC é o número total de subgrafos encontrados. Estes se dividem em quatro tipos: buracos com 7 ou mais vértices, anti-buracos com 7 ou mais vértices, buracos de 5 vértices (que é um grafo de Mycielski de alguma aresta) e subgrafos gerados a partir de operações de Mycielski, conforme definida no capítulo 2. O número de subgrafos de cada uma dessas classes foi contado e estão nas colunas $C7+$, $\bar{C}7+$, $C5$ e myc .

Nome	Coloração			Subgrafos					
	$\chi_C(G)$	$\chi_H(G)$	$\chi(G)$	SA	SC	$C7+$	$\bar{C}7+$	$C5$	myc
DSJC125.1	4	4	5	0	6937	6914	0	0	23
DSJC125.5	9	9	?	0	0	0	0	0	0
DSJ125.9	27	27	?	0	5924	0	5924	0	0
DSJ250.1	4	4	8	0	9669	4067	0	0	5602
DSJ250.5	8	9	?	0	130	0	0	0	130
DSJ250.9	34	34	?	0	3299	0	3299	0	0
queen5.5	5	5	5	1	16	0	0	0	16
queen6.6	4	5	7	0	1346	0	0	0	1346
queen7.7	5	5	7	0	163	0	0	0	163
queen8.8	4	5	9	0	10097	0	0	0	10097
queen8.12	4	5	12	0	29732	0	0	0	29732
queen9.9	5	5	10	0	345	0	0	0	345
queen10.10	4	5	10	0	38127	0	0	0	38127
queen11.11	5	5	11	0	521	0	0	0	521
FPSOL2.i.1	55	55	65	0	0	0	0	0	0
FPSOL2.i.2	29	29	30	0	0	0	0	0	0
FPSOL2.i.3	29	29	30	0	0	0	0	0	0
MUG88.1	2	3	4	22	136	111	0	25	0
MUG88.25	2	3	4	19	118	92	0	26	0
MUG100.1	2	3	4	10	183	162	0	21	0
MUG100.25	2	3	4	14	129	110	0	19	0

Nome	Números Cromáticos			Subgrafos					
	$\chi_C(G)$	$\chi_H(G)$	$\chi(G)$	SA	SC	$C7+$	$\bar{C}7+$	$C5$	myc
myciel3	2	4	4	1	30	0	0	29	1
myciel4	2	5	5	1	486	0	0	414	72
myciel5	2	6	6	1	6308	0	0	4088	2220
myciel6	2	7	7	1	83574	0	0	33682	49892
FullIns1.3	3	4	4	6	264	158	0	100	6
FullIns1.4	3	5	5	6	8395	4244	0	3721	430
FullIns1.5	3	6	6	6	188613	84028	0	85572	19013
FullIns2.3	4	5	5	1	418	402	0	0	16
FullIns2.4	4	6	6	1	27835	27070	0	0	765
FullIns3.3	5	6	6	1	839	828	0	0	11
FullIns4.3	6	7	7	1	3336	3321	0	0	15
FullIns5.3	7	8	8	1	7877	7858	0	0	19
Insertions1.4	2	3	4	564	3060	3060	0	0	0
Insertions1.5	2	3	?	6	67955	67955	0	0	0
Insertions2.3	2	3	4	18	217	217	0	0	0
Insertions2.4	2	3	?	87	12834	12834	0	0	0
Insertions3.3	2	3	4	24	414	414	0	0	0
Insertions3.4	2	3	?	1	38418	38418	0	0	0
Insertions4.3	2	3	4	15	707	707	0	0	0

Tabela 3. Detalhamento dos subgrfos identificados

6.4 Análise dos resultados

Se $LSC \neq LIC$, o processo foi abreviado, o que pode acontecer por dois motivos. O primeiro motivo ocorre quando $\lceil LSC \rceil = \lceil LIC \rceil$, o que implica o uso da propriedade 20 para determinar $\lceil \chi(G) \rceil$ sem resolver a geração de colunas até o fim. Este caso ocorreu particularmente na classe DSJC, onde o processo foi abreviado gerando limites inferiores melhores que os que aqueles conhecidos. Mais especificamente para o problema DSJC250.5 o limite inferior encontrado foi 26, bem superior ao antes conhecido 14. O segundo caso

para uma parada abreviada ocorre quando o limite de tempo foi atingido. Nesse caso, é fornecido, ainda, um limite inferior que é o maior entre $\chi_H(G)$ e o limite inferior calculado iterativamente.

É importante frisar que para os grafos FPSOL, o auxílio do pré-processamento permitiu uma retirada significativa de vértices, fazendo com que as dimensões do problema se tornassem tratáveis dentro do tempo estimado. Para essa classe de grafos a solução ótima foi encontrada em todas as instâncias, apesar da diferença entre $\omega(G)$ e $\chi(G)$.

Para algumas instâncias (MUGs e Insertions) a adição das restrições no modelo não melhora o limite inferior dado pela maior coloração de um subgrafo conhecido, embora em algumas instâncias a adição dessas restrições causa $LI > \chi_F(G)$.

Os grafos da classe myciel (grafos de Mycielski) e os grafos da classe FullIns foram resolvidos de forma ótima. Aqui cabe fazer um comentário que a coloração ótima de alguns dos grafos da classe FullIns estava em aberto.

Capítulo 7

Conclusões

Neste trabalho estudamos o problema de coloração de vértices de um grafo. Vimos inicialmente que existem várias maneiras de abordar o problema. Escolhemos estudar melhoras para o limite inferior para o problema, segundo a ótica da otimização matemática, mas também estudando outros subgrafos críticos, para extrair dessa combinação um melhor limite inferior.

Dentre as formulações citadas para o problema, escolhemos trabalhar com a Formulação por Conjuntos Independentes Maximais (CIM) e com a Formulação por Vértices Representantes (VR). A razão desta escolha se devia aos bons limites inferiores apresentados pela formulação CIM e por um conhecimento prévio das facetadas do poliedro da formulação VR, podendo-se assim apresentar uma solução que combinasse as duas qualidades. As técnicas utilizadas para resolver as formulações escolhidas foram brevemente recapituladas.

Relações entre as duas formulações foram mostradas e verificamos que existe uma equivalência entre CIM e uma decomposição Dantzig-Wolfe de VR. Então, classes de facetadas de VR foram adaptadas para CIM gerando desigualdades válidas para esta formulação. Além disso, a partir de subgrafos críticos, foram apresentadas algumas condições necessárias ou suficientes para que essas desigualdades definam facetadas para o politopo associado a CIM.

Um algoritmo de geração de linhas e colunas foi proposto para determinar um limite inferior para $\chi(G)$. O algoritmo foi implementado e testado para instâncias da DIMACS, onde $\omega(G) < \chi(G)$. Em geral, pode-se observar que o limite inferior encontrado foi o melhor entre o número cromático fracionário de G , $\chi_F(G)$, e a maior coloração dos

subgrafos identificados em G . Algumas instâncias onde, até então, não se conhecia o número cromático foram resolvidas em sua otimalidade. É o caso das instâncias FullIns.

Finalmente, foi apresentado um modo de fornecer um limite inferior para $\chi(G)$ sem que seja necessária a completa resolução da geração de colunas, possivelmente abreviando o término do processo. Essa estratégia se mostrou bastante eficiente nos grafos aleatórios da classe DSJC, permitindo determinar limites inferiores significativamente melhores que os conhecidos, em tempo aceitável.

Referências Bibliográficas

- [1] Bazaraa, M.S., Jarvis, J. J. e Sherali, H. D., *Linear Programming and Network Flows*, Wiley, 2a. edição, 1990.
- [2] Bondy, J.A. e Murty, U.S.R., *Graph Theory with Applications*. North-Holland, 1976.
- [3] Byskov, J. M., Enumerating Maximal Independent Sets with Applications to Graph Colouring, *Operations Research Letters*, 32(6): 547–556, 2004.
- [4] Campêlo, M., Corrêa, R. e Campos, V., On the asymmetric representatives formulation the vertex coloring problem. *Submitted to Discrete Applied Math.*, 2005.
- [5] Campêlo, M., Corrêa, R. e Frota, Y., Cliques, Holes and the Vertex Coloring Polytope. *Information Processing Letters*, 89:159-164, 2003.
- [6] Campêlo, M., Corrêa, R. e Frota, Y., Cliques, Holes and Lower Bounds for the Vertex Coloring Problem. In: *The 18th International Symposium on Mathematical Programming - ISMP 2003*, p.112, 2003.
- [7] Carraghan, R. e Pardalos, P., An Exact Algorithm for the Maximum Clique Problem. *Operations Research Letters*, 9:375–382, 1990.
- [8] Campos, V., *Estudo do Polítipo e dos Limites Inferiores Gerados pela Formulação de Coloração dos Representantes*. Dissertação de Mestrado, Universidade Federal do Ceará, 2005.
- [9] Chow, F. e Hennessy, J., The priority-based coloring approach to register allocation. *ACM Transactions on Programming Languages and Systems*, 12(4):501–536, 1990.
- [10] Chudnovsky, M., Robertson, N., Seymour, P. e Thomas, R., Progress on perfect graphs. *Mathematical Programming Series B*, 97(1-2):405–422, 2003.

- [11] Coll, P., Marenco, J., Díaz, I. e Zabala, P., Facets of the graph coloring polytope. *Annals of Operations Research*, 116(1–4):79-90, 2002.
- [12] Cornuéjols, G. e Sassano, A., On the 0,1 Facets of the Set Covering Polytope. *Math. Programming*, 43:45–55, 1989.
- [13] Díaz, I. e Zabala, P., A polyhedral approach for graph coloring. *Brazilian Symposium on Graphs, Algorithms and Combinatorics - GRACO 2001, Electronic Notes in Discrete Mathematics*, Vol. 7, 2001.
- [14] Ferreira, C. e Wakabayashi, Y., *Combinatória Poliédrica e Planos-de-Corte Faciais*. X Escola de Computação, 1996.
- [15] Figueiredo, R., *O poliedro das orientações acíclicas de um grafo sob restrições e suas aplicações*. Tese de Doutorado, Universidade Federal do Rio de Janeiro, 2005.
- [16] Figueiredo, R., Barbosa, V., Maculan, N. e Souza, C., New 0-1 integer formulations of the graph coloring problem. *Proceedings of XI CLAIO*, 2002.
- [17] Frota, Y., *Estudo Computacional de Algoritmos Exatos para Coloração de Grafos*. Dissertação de Mestrado, Universidade Federal do Ceará, 2003.
- [18] Gamst, A., Some lower bounds for a class of frequency assignment problems. *IEEE Transactions on Vehicular Technology*, 35(1):8–14, 1986.
- [19] Garey, M. e Johnson, D., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1978.
- [20] Hansen, P., Labbé, M. e Schindl, D., Set Covering and Packing Formulations of Graph Coloring: algorithms and first polyhedral results. *Manuscrito*, 2004.
- [21] Herrman e F., Hertz, A., Finding the chromatic number by means of critical graphs. *J. Exp. Algorithmics* 7, 2002.
- [22] Hertz, A. e Werra, D., Using Tabu Search Techniques for Graph Coloring. *Computing*, 39:345-351, 1987.
- [23] Hoffman, K. L. e Padberg, M., Solving airline crew scheduling problems by branch-and-cut. *Manage. Sci.*, 39(6):657–682, 1993

- [24] Johnson, Aragon, McGeoch, e Schevon, Optimization by Simulated Annealing: An Experimental Evaluation; Part II, Graph Coloring and Number Partitioning. *Operations Research*, 31, 378–406 (1991).
- [25] Johnson, D.S. e Trick, M.A. editores. Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, volume 26 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science. *American Mathematical Society*, 1996.
- [26] Larsen, M., Propp, J. e Ullman, D., The fractional chromatic number of a graph and a construction of Mycielsky. *Journal of Graph Theory*, 19(3):411–416, 1995.
- [27] Lucet, C., Mendes, F., Moukrim, A., Pre-processing and Linear-Decomposition Algorithm to Solve the k-Colorability Problem. *Experimental and Efficient Algorithms, Third International Workshop, WEA 3059:315-325*, 2004.
- [28] Lund, C. e Yannakakis, M., On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.
- [29] Mehrotra, A. e Trick, M., A Column Generation Approach for Graph Colouring. *INFORMS Journal in Computing*, 8(4): 344 - 354, 1996.
- [30] Mycielski, J., Sur le coloriage des graphes. *Colloq. Math* 3:161–162, 1955.
- [31] Nemhauser, G. e Wolsey, L., *Integer and Combinatorial Optimization*. Wiley, 1988.
- [32] Saad, Y., *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, MA, USA, 1996.
- [33] Sassano, A., On the facial structure of the set covering polytope. *Math. Programming*, 44:181–202, 1989.
- [34] Wolsey, L.A., *Integer Programming*. Wiley-Interscience Publication, 1998.
- [35] Dantzig, G.B. e Wolfe, P., Decomposition Principle for Linear Programs, *Operations Research*, 8: 101-111, 1960.
- [36] Werra, D. de, An introduction to timetabling. *European Journal of Operations Research*, 19:151–162, 1985.