



**UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**SOBRE PROBLEMAS DE LISTA COLORAÇÃO E A PROPRIEDADE DE
SELECIONABILIDADE EM GRAFOS**

SIMONE INGRID MONTEIRO GAMA

Abril de 2016

Manaus - AM

SIMONE INGRID MONTEIRO GAMA

SOBRE PROBLEMAS DE LISTA COLORAÇÃO E A PROPRIEDADE DE
SELECIONABILIDADE EM GRAFOS

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas (PPGI/IComp, UFAM) como parte dos requisitos necessários à obtenção do título de Mestre em Informática .

Orientadora: Rosiane de Freitas Rodrigues, D.Sc.

Abril de 2016

Manaus - AM

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

G184s Gama, Simone Ingrid Monteiro
Sobre problemas de lista coloração e a propriedade de
selecionabilidade em grafos / Simone Ingrid Monteiro Gama. 2016
81 f.: il. color; 31 cm.

Orientadora: Rosiane de Freitas Rodrigues
Dissertação (Mestrado em Informática) - Universidade Federal do
Amazonas.

1. coloração em grafos. 2. complexidade computacional. 3. lista
coloração. 4. selecionabilidade em grafos. 5. (gamma, mu)-
coloração. I. Rodrigues, Rosiane de Freitas II. Universidade Federal
do Amazonas III. Título



PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO DE COMPUTAÇÃO



UFAM

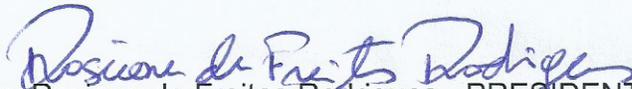
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

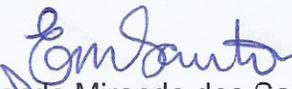
FOLHA DE APROVAÇÃO

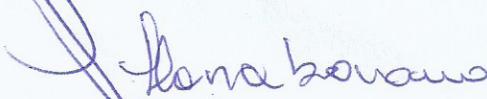
"Sobre Problemas de Lista Coloração e a Propriedade de
Selecionabilidade em Grafos"

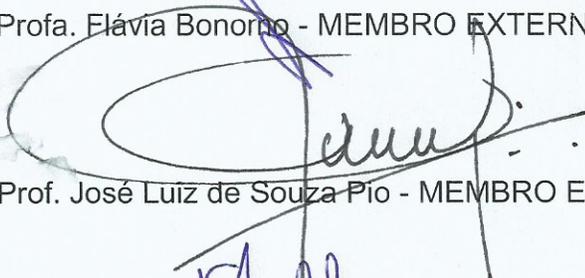
SIMONE INGRID MONTEIRO GAMA

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos
Professores:


Prof. Rosiane de Freitas Rodrigues - PRESIDENTE


Prof. Eulanda Miranda dos Santos - MEMBRO INTERNO


Prof. Flávia Bonorino - MEMBRO EXTERNO


Prof. José Luiz de Souza Pio - MEMBRO EXTERNO


Prof. Mário Salvatierra Junior - MEMBRO EXTERNO

Manaus, 19 de Abril de 2016

*Ouçã conselhos e aceite instruções,
e acabará sendo sábio.
Provérbios 19:20*

Agradecimentos

Agradeço primeiramente a Deus, hoje e sempre, pois tenho certeza que Ele me resgatou, me permitiu mais essa benção e estive sempre ao meu lado em todos os momentos.

A minha orientadora, Rosiane de Freitas Rodrigues, pela oportunidade de enriquecimento profissional e pessoal, dedicação, lições e pela competência com que sempre orientou esta pesquisa. Obrigada professora por acreditar em mim.

Ao meu professor, Mário Salvatierra, pela ajuda fundamental no entendimento das provas matemáticas e pela paciência, claro.

A minha família pelo amor, incentivo e apoio incondicional.

Aos meus amigos e companheiros de estudos, que me ajudaram diretamente e indiretamente neste trabalho e aos professores e pesquisadores com quem tive a oportunidade de estudar e que deram valiosas sugestões que contribuíram para o enriquecimento deste trabalho. Devo destacar aqui Rainer, Bruno e Raphael, vocês foram fundamentais na minha caminhada do mestrado, bem como o companheirismo de vocês.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo suporte financeiro dado na caminhada e sucesso deste trabalho.

Resumo da Dissertação apresentada ao PPGI/IComp/UFAM como parte dos requisitos necessários para a obtenção do grau de Mestre em Informática.

SOBRE PROBLEMAS DE LISTA COLORAÇÃO E A PROPRIEDADE DE
SELECIONABILIDADE EM GRAFOS

SIMONE INGRID MONTEIRO GAMA

Abril/2016

Orientadora: Profa. Rosiane de Freitas Rodrigues, D.Sc.

Nesta dissertação, o problema da lista coloração e a propriedade da selecionabilidade em grafos são abordados. Lista coloração é uma generalização do problema da coloração de vértices em grafos, e tal como este problema clássico, consiste em colorir um grafo simples de modo que vértices adjacentes possuam cores distintas, mas, respeitando-se a restrição adicional de que cada vértice possui uma lista restrita de possíveis cores a serem usadas. Tal problema possui algumas variações, como a (γ, μ) -coloração, que envolve listas sequenciais com limite inferior e superior conhecidos. A k -selecionabilidade consiste em determinar o menor tamanho de lista k para o qual sempre há uma lista coloração, seja qual for a distribuição de lista no grafo. Nesta dissertação, se investigou a correlação entre a propriedade da k -selecionabilidade e a (γ, μ) -coloração, sendo definida a propriedade de k - (γ, μ) -selecionabilidade. Com isto, foram também estudadas algumas técnicas de prova em selecionabilidade, aplicadas para se determinar a k - (γ, μ) -selecionabilidade para classes específicas de grafos, como a técnica de degeneração em grafos, usada para provar que grafos periplanares são 3- (γ, μ) -selecionáveis e a técnica de *Marshal Hall*, usada para provar que grafos bipartidos completos são 2- (γ, μ) -selecionáveis. O resultado mais geral, obtido através de uma prova formal, consistiu em determinar que se um grafo é k -colorível, então tal grafo também é k - (γ, μ) -selecionável, deixando de ser Π_2^P -completo para ser NP -completo. Adicionalmente, foram estudados e implementados alguns algoritmos para determinar a lista coloração e k -selecionabilidade em grafos simples gerais.

Palavras-chave: coloração em grafos, complexidade computacional, lista coloração, selecionabilidade em grafos, (γ, μ) -coloração, teoria dos grafos.

Abstract of Dissertation presented to PPGI/IComp/UFAM as a partial fulfillment of the requirements for the degree of Master in Informatics.

A STUDY ON THE PROBLEM OF LIST COLORING AND PROPERTY
CHOOSABILITY ON GRAPHS

SIMONE INGRID MONTEIRO GAMA

April/2016

Advisor: Profa. Rosiane de Freitas Rodrigues, D.Sc.

In this work, the list coloring problem and choosability property in graphs are investigated. List coloring is a generalization of the vertex coloring problem in graph, and as this classic problem is to color a simple graph so that adjacent vertices have different colors, but respecting the additional constraint that each vertex has a list of possible colors to be used. This problem has some variations as the (γ, μ) -coloring, which involves sequential lists with lower and upper bounds known. The k -choosability is to determine the smallest size list k for which there is always a valid list coloring, whatever the distribution of the list in the graph. Thus, we investigated the correlation between k -choosability and (γ, μ) -coloring, proposing the k - (γ, μ) -choosability property. With this, we also analysed some proof techniques in choosability, applied to determine k - (γ, μ) -choosability for specific graphs are 3- (γ, μ) -choosable and the technique of *Marshal Hall*, applied to prove that complete bipartite graphs are 2- (γ, μ) -choosable. The most general result, obtained through a relatively simple formal proof, consisted to determine if a graph is k -colorable, then this graph is also k - (γ, μ) -choosable, leaving to be Π_2^P -complete for NP -complete. Additionally, it was studied and implemented some algorithms to determine the list coloring and k -choosability for simple general graphs.

Keywords: choosability in graphs, computational complexity, graph coloring, graph theory, list coloring, (γ, μ) -coloring.

Sumário

Lista de Figuras	x
Lista de Tabelas	xiii
1 Introdução	1
1.1 Objetivos	4
1.2 Justificativa	4
1.3 Organização do trabalho	7
2 Fundamentos Teóricos	8
2.1 Teoria dos Grafos	8
2.1.1 Conceitos Básicos	8
2.1.2 Problemas de Coloração em Grafos	13
2.2 Complexidade Computacional	15
2.3 Considerações do capítulo	17
3 O Problema da Lista Coloração em Grafos e variações	18
3.1 Definições	18
3.2 (γ, μ) -coloração e μ -coloração	19
3.3 Algoritmos para lista coloração	22
3.3.1 Algoritmo para gerar todas as soluções para lista coloração	22
3.3.2 Algoritmo para determinar uma Lista Coloração de um grafo	25
3.3.3 Heurística gulosa DSATUR para lista coloração	28
3.3.4 Algoritmo para reconhecimento de uma solução em lista coloração	31
3.4 Considerações do capítulo	32

4	A Propriedade da Seleccionabilidade em Grafos	34
4.1	Definições	34
4.2	Técnicas de provas em seleccionabilidade	41
4.2.1	Técnica de degeneração em grafos	41
4.2.2	Técnica do sistema dos representantes distintos	42
4.2.3	Técnica de Thomassen	43
4.2.4	Técnicas que envolvem digrafos	45
4.3	Resultados em seleccionabilidade para classes específicas de grafos	46
4.3.1	Seleccionabilidade em grafos bipartidos e multipartidos	46
4.3.2	Seleccionabilidade em grafos planares	47
4.3.3	Seleccionabilidade versus número cromático	50
4.4	Algoritmo para determinar a <i>k-seleccionabilidade</i>	51
4.5	Considerações do capítulo	55
5	Contribuições	56
5.1	Definição da propriedade da <i>k-(γ, μ)-seleccionabilidade</i>	56
5.2	Aplicação das técnicas de provas de seleccionabilidade em <i>k-(γ, μ)-seleccionabilidade</i>	57
5.2.1	<i>k-(γ, μ)-seleccionabilidade</i> em grafos periplanares	57
5.2.2	<i>k-(γ, μ)-seleccionabilidade</i> em grafos bipartido completo	59
5.3	Grafos <i>k-coloríveis</i> são <i>k-(γ, μ)-seleccionáveis</i>	64
5.4	Algoritmos para determinar a <i>k-(γ, μ)-seleccionabilidade</i>	66
5.5	Considerações do capítulo	72
6	Considerações Finais	73
	Referências	75

Lista de Figuras

1.1	Coloração do Mapa do Amazonas representado por um grafo planar . . .	2
1.2	Exemplo do Quadrado Latino 3×3 e sua representação em forma de grafo.	3
1.3	Exemplo do Quadrado Latino 3×3 com listas de cores de tamanho 2. . .	4
1.4	Alocação de canais em redes sem fio modelado como lista coloração. . . .	5
1.5	Aplicação da lista coloração em atividades econômicas de municípios . .	6
2.1	Representação gráfica de um grafo simples com 7 vértices e 10 arestas. . .	9
2.2	Exemplos de subgrafos H de um grafo G	9
2.3	Grafo completo com 4 vértices.	9
2.4	Exemplo de grafo periplanar.	10
2.5	Grafo árvore de ordem seis.	11
2.6	Grafo bipartido, grafo caminho e grafo ciclo	12
2.7	Grafos de Finck, cujo o conjunto de vértices são particionados em uma clique, um conjunto independente e um C_5 (no caso da Figura 2.7(b)). . .	13
2.8	Coloração própria do grafo da Figura 2.1.	14
2.9	Coloração mínima dos vértices da Figura 2.1 sendo que o seu número cromático equivale a 3 (três).	14
2.10	Coloração de arestas da Figura 2.3.	15
3.1	Lista coloração de um grafo	19
3.2	μ -coloração e (γ, μ) -coloração	20
3.3	Esquema de generalizações entre problemas de coloração em grafos. . . .	20
3.4	Árvore de execução do algoritmo exato de lista coloração	24
3.5	Exemplo de soluções da árvore de execução do algoritmo exato	24
3.6	Soluções encontradas do algoritmo	25

3.7	Grafo com lista de cores associadas para execução do Algoritmo	27
3.8	Árvore de execução do algoritmo para determinar uma lista coloração de um grafo	27
3.9	Exemplo de execução do Algoritmo DSATUR.	29
3.10	Execução do Algoritmo L-DSATUR	31
3.11	Exemplo de solução em lista coloração	32
4.1	Exemplo de um grafo bipartido $K_{3,3}$ que não é 2-seleccionável.	35
4.2	θ grafos	35
4.3	$\theta_{2,2,2k}$ grafos	36
4.4	Exemplo de um Grafo $\theta_{2,2,2k}$ rotulado para a prova.	36
4.5	Cinco tipos de Grafos para o teorema	38
4.6	Grafo $C_1 \cup P_1$ para a prova.	38
4.7	Grafo do Caso 1 e 2 $C_1 \cup P_1 \cup P_2$	39
4.8	Grafo do Caso 3 e 4 $C_1 \cup P_1 \cup P_2$	39
4.9	Grafo do Caso 6 $C_1 \cup P_1 \cup P_2$	39
4.10	Grafo onde as extremidades de P_2 são b e b'	40
4.11	Exemplo de listas de cores para os cinco tipos de grafos que não são 2-seleccionável.	40
4.12	Grafo Triangulado e <i>Quase-Triangulado</i>	43
4.13	Grafo <i>Quase-Triangular</i> com corda	44
4.14	Grafo <i>Quase-Triangular</i> sem corda	45
4.15	Exemplo de Execução do Algoritmo 1	53
4.16	Exemplo de solução encontrada com a execução do algoritmo para seleccionabilidade em grafos.	54
5.1	Grafo K_3 em que o $\chi_{\ell,(\gamma,\mu)} = 3$	57
5.2	Grafo periplanar maximal	58
5.3	Grafo $K_{n,n}$ com partições com listas de cores iguais.	59
5.4	Exemplo de um grafo $K_{3,3}$ para a prova	60
5.5	Subgrafo de um $K_{n,n}$. Coloração feita nos vértices v_{n+1} e w_{n+1}	60
5.6	Grafo $K_{n,n}$ com partições com listas de cores diferentes.	61

5.7	Grafo $K_{4,4}$. Cada vértice foi colorido com uma cor distinta c_i tal que $L(v) = S_i$, cumprindo o Teorema 4.3.	61
5.8	$K_{n,n}$ com os conjuntos S_1, S_2, \dots, S_r	62
5.9	$K_{n,n}$ com os conjuntos S_1, S_2, \dots, S_r colorido	63
5.10	Conjuntos E_1, E_2, \dots, E_t e F_1, F_2, \dots, F_s	63
5.11	Uma k - (γ, μ) -coloração de um grafo, que é 3-colorível, portanto é 3- (γ, μ) -selecionável.	65
5.12	Exemplo de execução do algoritmo.	68
5.13	Soluções encontradas para o grafo da Figura 5.12	68
5.14	Gráficos de comparação da geração de listas de cores	71

Lista de Tabelas

3.1	Tabela de comparação da complexidade computacional para coloração, lista coloração e suas variações.	21
4.1	Resultados de selecionabilidade em grafos que utilizam como prova a técnica de degeneração em grafos.	41
4.2	Resultados de selecionabilidade em grafos que utilizam a técnica de prova do sistemas dos representantes distintos.	43
4.3	Resultados da k -selecionabilidade para grafos multipartidos.	47
4.4	Resultados da não k -selecionabilidade para grafos planares.	48
4.5	Resultados da k -selecionabilidade para grafos planares.	49
4.6	Tabela de comparação dos resultados existentes na literatura para k -coloração e k -selecionabilidade em diversas classes de grafos.	50
5.1	Quantidade de listas de cores geradas pelos algoritmos da k -selecionabilidade (χ_ℓ) e k - (γ, μ) -selecionabilidade ($\chi_{\ell, (\gamma, \mu)}$).	71

Capítulo 1

Introdução

Coloração em grafos é um dos problemas mais antigos e bem conhecidos da teoria dos grafos. Muitos problemas de interesse teórico e prático podem ser modelados como problema de coloração em grafos, possuindo uma infinidade de aplicações.

A origem de tal problema teórico se deu com a questão de se colorir um mapa, ou seja, de se diferenciar regiões adjacentes de um mapa usando o menor número de cores possível (cada região deve ser colorida com uma cor). A proposta, aparentemente simples, atraiu a atenção de muitos matemáticos, dando origem a Conjectura das Quatro Cores (do inglês, *Four Colour Conjecture*)[52], que afirmava que para se colorir um mapa, bastam apenas 4 cores. Foram realizadas várias tentativas para resolver tal conjectura, ficando sem solução por 120 anos e sendo finalmente provada em 1977 por Appel, Haken e Koch [36]. A prova foi elaborada com a ajuda de um computador para verificar centenas de casos especiais de coloração. Até o momento não se encontrou uma prova puramente matemática, sendo a prova mais atual de 1996 [56], onde os autores refinaram a prova anterior, mas ainda se utilizaram de um computador para esta finalidade. A coloração mínima de um grafo é um problema conhecido ser NP-Difícil [35].

Em termos de teoria dos grafos, esse problema pode ser representado como o problema de **coloração de vértices** de um grafo planar, de tal modo que os vértices representam as regiões do mapa e as arestas relacionam as regiões vizinhas que devem ser coloridas com cores diferentes. Um exemplo seria os municípios do mapa do Estado do Amazonas, na Figura 1.1, onde os vértices do grafo representam os municípios e as arestas representam as regiões vizinhas.

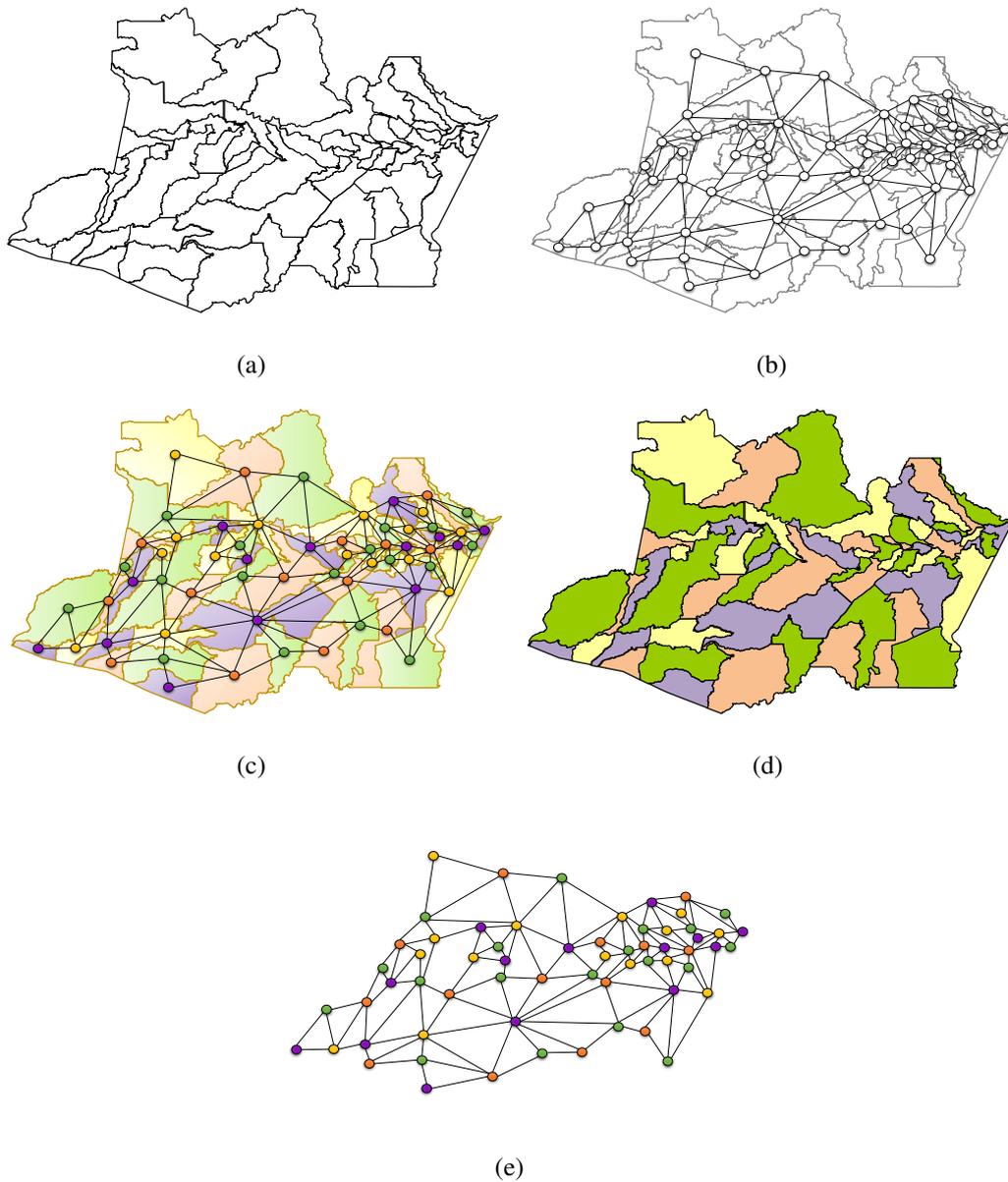


Figura 1.1: Um exemplo de coloração do mapa do Estado do Amazonas dividido em municípios. Em (a) o mapa do Estado do Amazonas. Em (b), o mapa representado por um grafo planar. Em (c), o mesmo mapa e seu respectivo grafo colorido. Em (d), o mapa colorido. E por fim, em (e) o respectivo grafo do mapa do Amazonas colorido com 4 cores.

Uma variação interessante do problema clássico de se colorir corretamente os vértices de um grafo surge quando são impostas restrições adicionais sobre as cores disponíveis para cada vértice. Uma dessas variações tem recebido uma atenção considerável, e que levou a várias conjecturas [59], teoremas e resultados interessantes, conhecido como o Problema da Lista Coloração (do inglês, *List Coloring Problem*). Neste caso, a cada vértice de um grafo é dada uma lista de cores possíveis, e aos vértices são atribuídas

cores de suas listas, de tal forma que vértices adjacentes recebam cores diferentes. Tal problema foi introduzido por Paul Erdős *et al.*, em 1979 [59] e, independentemente por Vadim Vizing em 1976 [73], sendo bem estudado desde então. Erdős *et al.* iniciaram os estudos em lista coloração e selecionabilidade em grafos baseando-se no Problema de Dinitz (do inglês, *Dinitz's Problem*), introduzido por Jeff Dinitz [63]. O problema é o que se segue: considere n^2 células arranjadas em um $(n \times n)$ -quadrado, e seja (i, j) denotado por uma célula na linha i e coluna j . Suponha que toda célula (i, j) tenha um conjunto $C(i, j)$ de n cores, e se possível, deseja-se colorir toda a matriz, escolhendo para cada célula (i, j) de uma cor a partir do seu conjunto de $C(i, j)$ de tal modo que as cores em cada linha e em cada coluna sejam distintos?

A Figura 1.2 apresenta um exemplo do quadrado latino representado na forma de grafos. Na Figura 1.2(a) é apresentado um quadrado 3×3 com listas de cores (representadas por números naturais inteiros) associadas a cada célula. Na Figura 1.2(b), cada célula recebe uma cor, sem violar as restrições. Já na Figura 1.2(c), o quadrado 3×3 está representado na forma de grafo, onde cada vértice representa cada célula (i, j) e as arestas entre as células indica que uma célula está na mesma linha e mesma coluna a outra célula a qual é adjacente.

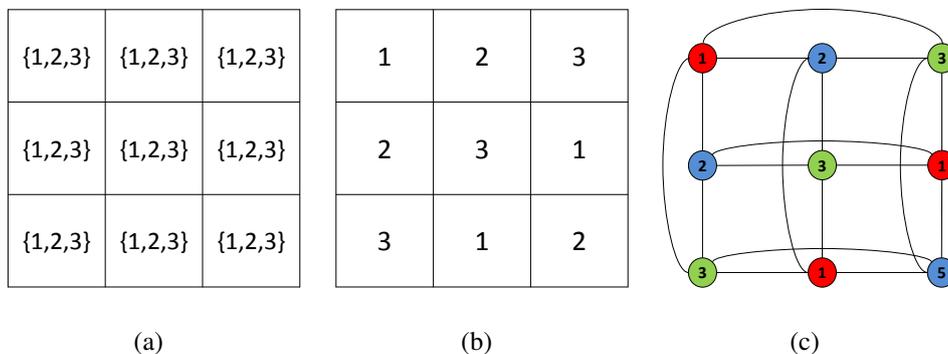


Figura 1.2: Exemplo do Quadrado Latino 3×3 e sua representação em forma de grafo.

Colorir uma matriz cuja quantidade de cores de cada célula $C(i, j)$ é igual ou maior que n é fácil. Agora, colorir a mesma matriz com uma quantidade menor que n cores para cada célula é mais difícil [63]. A Figura 1.3 apresenta um quadrado latino colorido, onde as listas de cores possuem tamanho menor que n . Nesse caso, é possível uma coloração do grafo resultante.

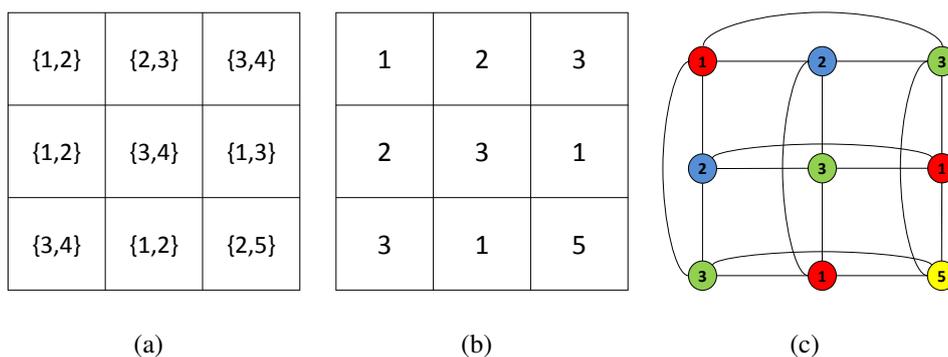


Figura 1.3: Exemplo do Quadrado Latino 3×3 com listas de cores de tamanho 2.

1.1 Objetivos

Neste trabalho deseja-se investigar o problema da lista coloração e suas variações, além da propriedade de selecionabilidade, correlacionando a selecionabilidade em grafos com a (γ, μ) -coloração. Os objetivos específicos são:

- Apresentar em detalhes as provas dos principais resultados e teoremas da propriedade de selecionabilidade em grafos.
- Correlacionar a propriedade de selecionabilidade com a (γ, μ) -coloração, uma variação da lista coloração, gerando assim a k - (γ, μ) -selecionabilidade (ou k - (γ, μ) -coloração).
- Aplicar algumas das técnicas de provas de selecionabilidade para provar a k - (γ, μ) -coloração em algumas classes de grafos.

1.2 Justificativa

O problema da lista coloração possui diversas variações e propriedades ainda não estudadas, principalmente em relação a propriedade de selecionabilidade em grafos. Apesar de ser um problema teórico, a lista coloração pode ser mapeada para inúmeros problemas reais, contribuindo fortemente para justificar o presente trabalho. Além disso, justifica-se pela oportunidade de associar uma das variações da lista coloração, a (γ, μ) -coloração, com uma propriedade muito interessante dessa coloração, a selecionabilidade, de forma

a verificar o comportamento desse tipo de coloração gerado por essa relação com os resultados existentes na literatura, além da sua aplicabilidade em diversos problemas, como alguns listados a seguir:

Alocação de canais em redes sem fio

O problema da lista coloração pode ser modelado como distribuição de canais para cada conexão de diferentes transmissores em uma rede sem fio, utilizando uma faixa limitada do espectro eletromagnético e evitando ou minimizando interferências. As estações de rádios são representados pelos vértices, as arestas representam as estações que interferem entre si, os domínios de frequência como listas de cores possíveis e os canais como cores atribuídas a cada vértice.

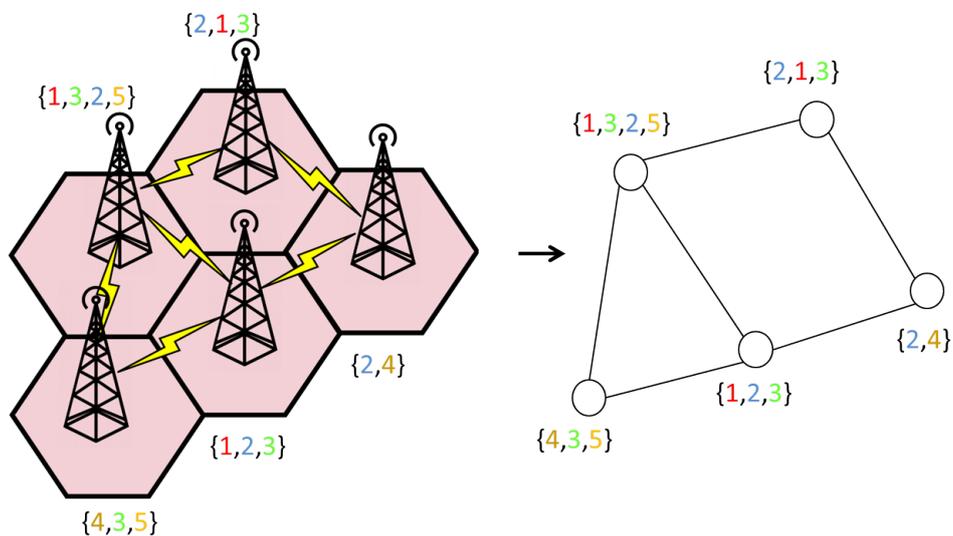


Figura 1.4: Alocação de canais em redes sem fio modelado como lista coloração.

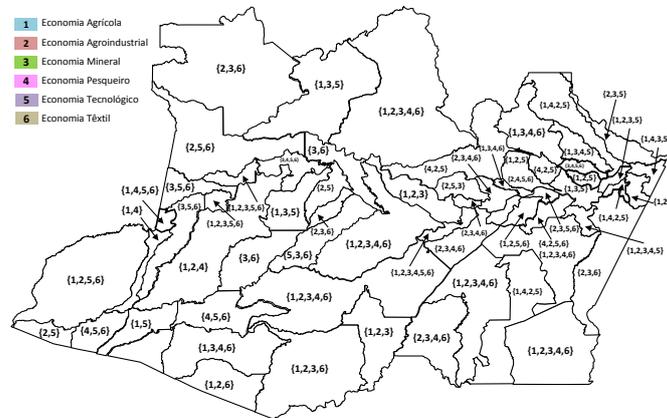
Atividades econômicas dos municípios de um estado

Um estado recebe recursos do governo federal para que os mesmos sejam distribuídos entre os seus municípios. Cada município recebe seu recurso econômico de acordo com uma classificação estabelecida pelo Manual da Receita Federal ou de acordo com um plano estabelecido pelos gestores do município em questão, com o objetivo de cumprir com a efetiva arrecadação do município, de acordo com a Lei de Responsabilidade Fiscal [11].

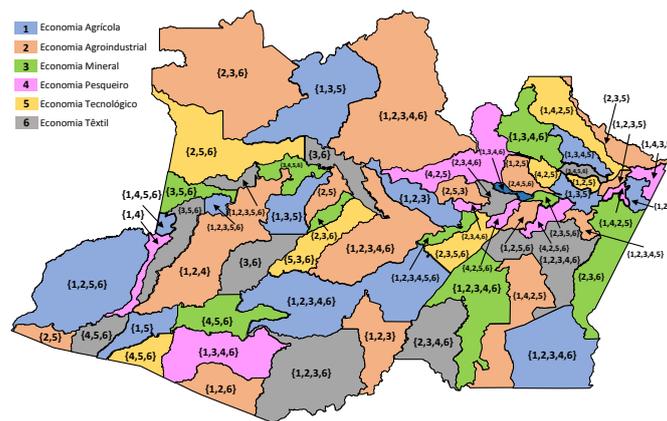
É necessário que cada município seja classificado de acordo com sua principal atividade econômica para o recebimento de repasse de verbas. Hipoteticamente, se uma

cidade possui uma atividade econômica de destaque, o município adjacente à esta cidade não pode receber a mesma classificação, pois de acordo com regras estabelecidas pelo governo federal, se uma cidade é caracterizada por sua atividade econômica no campo tecnológico por exemplo, a população da cidade vizinha será beneficiada pelos mesmos recursos, com o objetivo de economizar nos repasses de verbas para cada cidade.

Esse problema pode ser modelado como lista coloração em grafos, onde a correlação será feita da seguinte forma: as cidades de um estado serão os vértices, as arestas representam as cidades que são adjacentes, as listas de cores de cada cidade representam os polos econômicos que cada cidade pode ter, e a cor selecionada para um vértice representa que a cidade recebe uma classificação econômica permitida.



(a)



(b)

Figura 1.5: Um exemplo de coloração do mapa do Estado do Amazonas dividido em municípios. Em (a) o mapa do Estado do Amazonas com suas respectivas cidades e seus respectivos pólos. Em (b) o mapa colorido, representando os pólos a qual cada cidade foi classificada.

A Figura 1.5(b) apresenta um exemplo dessa correlação. O mapa do estado do Amazonas na Figura 1.5(a) apresenta cada cidade com uma lista de classificação de polos econômicos, que será representada pela lista de cores. A Figura 1.5(b) apresenta o mapa do Amazonas já colorido com as cores de sua lista de cores, ou seja, cada cidade recebeu sua respectiva classificação econômica, de tal modo que duas cidades vizinhas não podem receber a mesma classificação, de acordo com as regras estabelecidas pelo governo federal.

1.3 Organização do trabalho

Com relação a organização, esta dissertação encontra-se dividida em 06 (seis) capítulos, que são: Neste capítulo é apresentada a introdução desta dissertação. No Capítulo 2 é apresentado o referencial teórico necessário para o entendimento da pesquisa realizada, onde são introduzidos os conceitos de Teoria dos Grafos e Complexidade Computacional. No Capítulo 3, é apresentado o problema da Lista Coloração, algumas variações, bem como uma tabela com os trabalhos correlatos comparando os resultados existentes na literatura entre essas colorações. É apresentado também os algoritmos relacionados a lista coloração. No Capítulo 4, é apresentado sobre a propriedade da selecionabilidade em grafos, as provas das principais propriedades e as principais técnicas de provas relacionadas a selecionabilidade são apresentadas, além de um algoritmo exato para essa propriedade. No Capítulo 5, é apresentado as principais contribuições relacionadas a $k-(\gamma, \mu)$ -selecionabilidade, aplicando as técnicas de provas apresentados no capítulo anterior, além dos algoritmos para $k-(\gamma, \mu)$ -selecionabilidade. No Capítulo 6, são feitas as considerações finais sobre os resultados desta dissertação diretas para trabalhos futuros.

Capítulo 2

Fundamentos Teóricos

Neste capítulo são apresentados as definições que formam a base para o desenvolvimento deste trabalho, envolvendo os principais conceitos em Teoria dos Grafos, Coloração em Grafos, e Complexidade computacional.

2.1 Teoria dos Grafos

Esta seção apresenta conceitos básicos em teoria dos grafos, bem como algumas de suas classes importantes. As terminologias e notações utilizadas no restante deste documento são introduzidas nesta seção e utilizaram os livros de Bondy e Murty [4], Jaime Szwarc-fiter [68] e Chartrand [24].

2.1.1 Conceitos Básicos

Um grafo $G = (V, E)$ é um conjunto finito não-vazio onde $V(G)$ é um conjunto de vértices e $E(G)$ é um conjunto de arestas [68]. Cada elemento e no conjunto E é um par (i, j) que indica que o vértice i é adjacente ao vértice j (ou seja, são adjacentes, e a aresta e incide em i e j). A representação gráfica de um grafo consiste em pontos distintos do plano associados a cada vértice e, para cada aresta (i, j) , um segmento de reta conectando os pontos correspondentes aos vértices i e j . Um exemplo de grafo é mostrado na Figura 2.1.

O número de vértices de um grafo G é dito ser a **ordem** de G e o número de arestas é o **tamanho** de G [24]. O grau de um vértice v do grafo equivale à quantidade de arestas que incidem em v . O grau máximo do grafo, denotado por $\Delta(G)$, é o valor do maior grau dentre todos os vértices de G . De maneira similar, o grau mínimo, denotado por $\delta(G)$, é

definido como o valor do menor grau de G . O conjunto de vizinhos de um vértice v de G é denotado por $N_G(v)$.

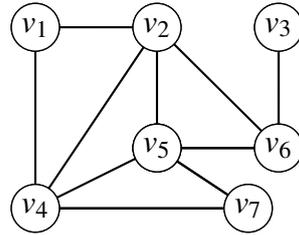


Figura 2.1: Representação gráfica de um grafo simples com 7 vértices e 10 arestas.

Definição 2.1 (Szwarcfiter, 1986 [68]). Um grafo H é dito ser um **subgrafo** de G se $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. Se além disso H possuir toda aresta (uw) de G tal que ambos os vértices estejam em $V(H)$, então H é o **subgrafo induzido** pelo subconjunto de vértices de H . Diz-se então que H induz G .

Um exemplo de subgrafos da Figura 2.1 é mostrado na Figura 2.2.

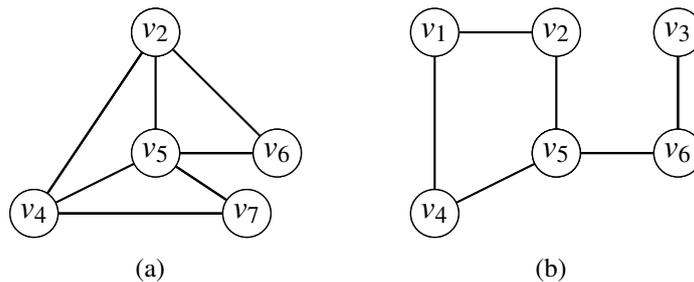


Figura 2.2: Exemplo de Subgrafos do grafo G da Figura 2.1. Em (a), H é um subgrafo induzido de G . Em (b), H é um subgrafo de G , mas não é induzido.

Definição 2.2. Um **grafo completo**, denotado por K_n , é aquele em que cada vértice é vizinho de todos os demais, ou seja, cada par de vértices é adjacente entre si.

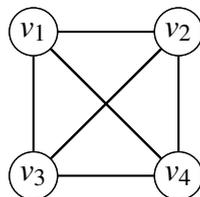


Figura 2.3: Grafo completo com 4 vértices.

O grau de todos os vértices de um grafo completo é exatamente igual a $n - 1$. A quantidade de arestas é dada então por $m = \frac{n(n-1)}{2}$. Um exemplo de K_4 é mostrado na Figura 2.3.

Definição 2.3. Um grafo é **conexo** se quaisquer que sejam os vértices distintos u e v de G , existe sempre um caminho que os une. Um grafo que não é conexo é chamado **desconexo**.

Definição 2.4. Um grafo que possui uma representação gráfica sem que as suas arestas se cruzem é chamado de grafo **planar**.

Um exemplo de grafo planar é mostrado na Figura 2.1. Um grafo planar divide o plano em regiões chamadas de **faces**. A face do grafo que é ilimitada é chamada de face externa enquanto as outras faces são chamadas de faces internas. Na Figura 2.1 temos cinco faces, sendo quatro faces internas e a face externa.

Definição 2.5 (Bondy [4]). Um grafo G é **periplanar** se o mesmo é planar e todos os vértices de G permanecem limitados na região exterior.

Um exemplo de grafo periplanar é visto na Figura 2.4. Os vértices do grafo periplanar ficam limitados apenas a região externa. Um grafo G é periplanar maximal se a inserção de uma aresta ao grafo o torna não periplanar.

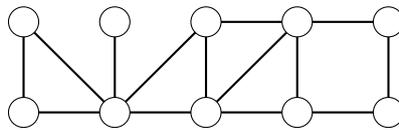


Figura 2.4: Exemplo de grafo periplanar.

Definição 2.6. Uma **clique** de um grafo $G = (V, E)$ é um subconjunto $V' \subseteq V$ cujo subgrafo induzido $G[V']$ é completo.

Os vértices v_4, v_5 e v_7 da Figura 2.1 são exemplos de vértices que pertencem a uma clique. Uma clique máxima de um grafo G é aquela com a maior cardinalidade dentre todas as cliques de G , e o número da clique $\omega(G)$ é a cardinalidade da clique máxima de G .

Definição 2.7. Um **conjunto estável** (ou **independente**) de um grafo $G = (V, E)$ é um subconjunto $V' \subseteq V$ em que nenhum par de vértices é adjacente, para qualquer vértice de V' .

Os vértices v_1, v_3 e v_5 da Figura 2.1 são exemplos de vértices que pertencem a um conjunto estável.

Definição 2.8. Um grafo é *split* se seu conjunto de vértices pode ser particionado em uma clique e em um conjunto estável.

Um grafo é *split* completo se todos os vértices de sua clique são adjacentes a todos os vértices de seu conjunto estável.

Definição 2.9. Um grafo k -partido é um grafo cujos vértices podem ser particionados em k conjuntos disjuntos, nos quais não há arestas entre vértices de um mesmo conjunto.

Se cada dois vértices de conjuntos diferentes são adjacentes, esse é um grafo k -partido completo. Um grafo 2-partido é conhecido como grafo bipartido. Logo, se cada dois vértices, um de cada conjunto disjunto são adjacentes, então têm-se um grafo bipartido completo. Um exemplo de um grafo bipartido completo $K_{3,3}$ é mostrado na Figura 2.6(a).

Definição 2.10 (Szwarcfiter, 1986 [68]). Uma **árvore** é um grafo conexo, acíclico e que possui $n - 1$ arestas.

Um exemplo de uma árvore é dado na Figura 2.5. Se um vértice v da árvore possui grau um, então v é uma folha. Toda árvore é um grafo bipartido e planar. Uma árvore com um nó interno e k folhas ou $K_{1,k}$ (grafo bipartido completo) é chamada de **estrela**. Uma estrela com 3 arestas é chamada de **garra** (do inglês, *claw*).

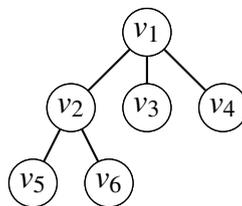


Figura 2.5: Grafo árvore de ordem seis.

Definição 2.11 (Szwarcfiter, 1986 [68]). Um **grafo caminho** é uma sequência de vértices $P = \{v_1, v_2, \dots, v_n\}$ de tal forma que dois vértices são adjacentes se forem consecutivos na sequência, e não adjacentes, caso contrário.

O comprimento ou tamanho de P , denotado por $|P|$, é o número de arestas de P , ou seja, $|P| = n - 1$. Um exemplo de caminho é mostrado na Figura 2.6(b).

Definição 2.12. Um **grafo ciclo** C_n (para $n \leq 3$) é um grafo simples cujo os vértices podem ser dispostos em uma sequência cíclica de tal forma que dois vértices são adjacentes se são consecutivos na sequência, e não adjacentes em caso contrário. O primeiro e último vértices coincidem.

Um exemplo de um ciclo C_4 é mostrado na Figura 2.6(c). O tamanho de um ciclo é o número de suas arestas. Um ciclo de tamanho k é chamado de k -ciclo. Um grafo G é **acíclico** se o mesmo não contém ciclo.

Definição 2.13 (Andreas, 1999 [41]). *Um grafo G é um **grafo cordal** se cada ciclo em G de tamanho pelo menos 4 tem uma corda. Uma corda é uma aresta que conecta dois vértices não adjacentes neste ciclo.*

A Figura 2.3 é um exemplo de um grafo cordal, pois a aresta $\{v_1, v_4\}$ representa uma corda do ciclo $\{v_1, v_2, v_4, v_3, v_1\}$.

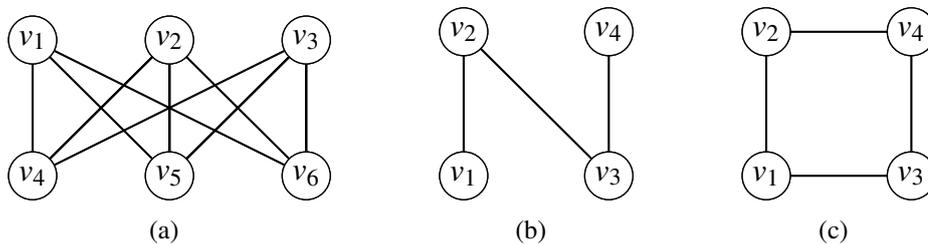


Figura 2.6: Exemplos de Grafos: (a) Grafo Bipartido Completo, (b) Grafo Caminho e (c) Grafo Ciclo

Definição 2.14 (Andreas, 1999 [41]). *Seja F uma família de **grafos proibidos**. Um grafo G é **F -livre** se G não contém algum subgrafo induzido que pertença à F*

Para grafos proibidos, um exemplo é um grafo **triângulo-livre** (ou K_3 -livre) é um grafo que não contém ciclos de tamanho 3 como subgrafo induzido. Um exemplo de um grafo triângulo-livre é a Figura 2.5, pois na mesma não contém ciclos de tamanho 3 ou mais, portanto é um triângulo-livre. De forma análoga, um grafo **livre de garra** (garra-livre ou $K_{1,3}$ -livre) é um grafo que não tem qualquer garra como subgrafo induzido. Um exemplo de grafo livre de garra é mostrado na Figura 2.6(c). Grafos que são livres de P_4 (ou P_4 -livre) e grafos livres de C_4 (C_4 -livre ou sem C_4) também são chamados de **cografos** [41].

Definição 2.15 (Bondy [4]). *Um grafo G é **k -degenerado** se cada subgrafo H de G tem um vértice de grau no máximo k .*

Outra de definição apresentada pelo autor: um grafo é k -degenerado se pode ser reduzido a um K_1 por repetidas deleções de seus vértices de grau no máximo k . Um exemplo

são grafos árvores, que são 1-degenerados. Grafos planares são 5-degenerados, pois possuem um vértice de até no máximo grau 5. Grafos periplanares são 2-degenerados [45]. Desde que todo grafo periplanar possui dois vértices não adjacentes de grau no máximo 2 e cada subgrafo de um grafo periplanar também é periplanar, então todo grafo periplanar é 2-degenerado [45].

Uma classe de grafos interessante que será abordada nos resultados da literatura em selecionabilidade são conhecidos como grafos de Finck [18], que são grafos cujo o conjunto de vértices podem ser particionados em uma clique K e um conjunto independente S , onde um dos vértices de S é adjacente a todos os vértices de K (veja Figura 2.7(a)) e grafos cujo conjunto de vértices pode ser particionado em uma clique K , um conjunto independente S e um C_5 tal que todo vértice de C_5 é adjacente a todo vértice de K e a nenhum vértice de S (veja Figura 2.7(b)).

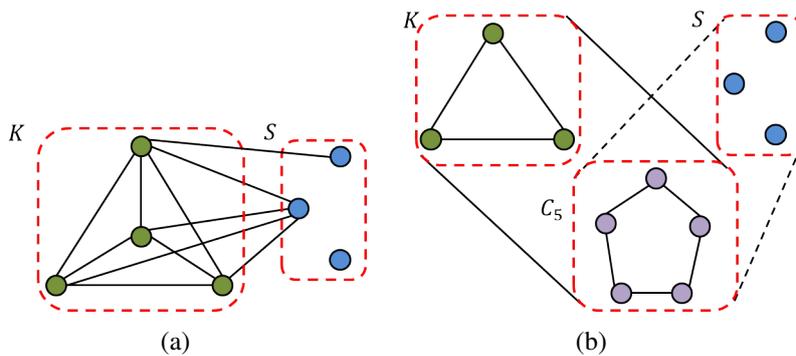


Figura 2.7: Grafos de Finck, cujo o conjunto de vértices são particionados em uma clique, um conjunto independente e um C_5 (no caso da Figura 2.7(b)).

2.1.2 Problemas de Coloração em Grafos

Esta seção apresenta os principais conceitos relacionados à coloração de grafos. O estudo de coloração em grafos foi iniciado com o problema das quatro cores, como mencionado no Capítulo 1.

Definição 2.16. *Uma **coloração própria** c de vértices em um grafo G é uma atribuição de cores aos vértices de G tal que vértices adjacentes possuem cores diferentes.*

As cores são representadas por números inteiros positivos $1, 2, \dots, k$. Então, uma coloração própria pode ser considerada como uma função $c : V(G) \rightarrow \mathbb{N}$ (onde \mathbb{N} é o conjunto de números inteiros positivos) tal que $c(v) \neq c(u)$ se u e v são vértices adjacentes em G .

Definição 2.17 (Bondy, 1982 [4]). *Um grafo G que é colorido com k cores possui uma k -coloração.*

O grau de saturação de um vértice $v \in V(G)$, é o número de vértices previamente coloridos e adjacentes a v . O vértice v_2 da Figura 2.8 possui grau de saturação 4, pois é adjacente a quatro vértices coloridos.

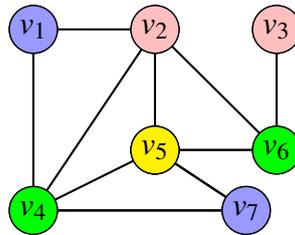


Figura 2.8: Coloração própria do grafo da Figura 2.1.

Definição 2.18 (Bondy, 1982 [4]). *Um grafo G é k -colorível se existe uma k coloração para G . O mínimo inteiro positivo para o qual G é k -colorível é chamado de **número cromático** de G e é denotado por $\chi(G)$ (lê-se chi de G).*

O número cromático de um grafo é também o número mínimo de conjuntos independentes em que o conjunto de vértices $V(G)$ é particionado. O objetivo é minimizar a quantidade de cores usadas, que equivale a determinar o número cromático do grafo, que é conhecido ser um problema NP-Difícil [35].

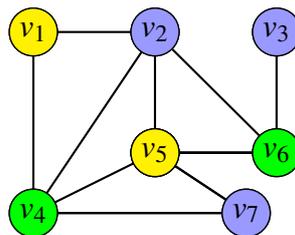


Figura 2.9: Coloração mínima dos vértices da Figura 2.1 sendo que o seu número cromático equivale a 3 (três).

Um grafo G com número cromático igual a k é um grafo k -cromático. Um grafo completo G com n vértices é n -cromático, uma vez que cada vértice de G é adjacente a todos os $n - 1$ vértices de G , serão necessárias n cores para colorir G [24].

Existem outras formas de coloração em grafos, como por exemplo, **coloração de arestas**, onde as cores são atribuídas as arestas de tal modo que arestas que compartilham o mesmo vértice não possuem as mesmas cores. Uma coloração de arestas com k cores é

chamada de ***k*-aresta-coloração**, um exemplo é mostrado na Figura 2.10. Outra forma de coloração é a **coloração total em grafos**, que é uma coloração de vértices e arestas, no sentido de que nenhum vértice adjacente, nenhuma aresta adjacente e sem que arestas e seus respectivos vértices compartilhem a mesma cor.

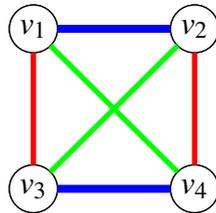


Figura 2.10: Coloração de arestas da Figura 2.3.

Existem inúmeras variações do problema de coloração em grafos, dentre algumas destacam-se: a **soma coloração**, onde o objetivo é minimizar a soma das cores atribuídas aos vértices de um grafo. Outra variação muito conhecida é a ***T*-coloração**, onde dado um conjunto T de inteiros não negativos com zero, onde uma coloração de G é dada de tal forma que $|c(u) - c(v)| \notin T$ para todo $(u, v) \in E(G)$, ou seja, a distância entre duas cores de vértices adjacentes não deve pertencer ao conjunto T [27]. Os vértices também podem ter restrições de quais cores os mesmos podem assumir, como a coloração por listas de cores, onde cada vértice possui uma lista de cores possíveis associadas à seus vértices e que podem ser usadas, e que é o objetivo deste trabalho. Mais detalhes de coloração por listas serão apresentados no Capítulo 3.

2.2 Complexidade Computacional

Um problema de otimização consiste em encontrar a melhor solução dentre um conjunto de conjuntos. A qualidade da solução é medida por meio de uma **função objetivo**, sendo que em um problema de minimização, a melhor solução será aquela de menor valor, e em um problema de maximização, a solução será a de maior valor [8]. Há muitas classificações possíveis para o problema de otimização, e algumas delas apresentam métodos exatos e de resolução eficientes. Outras levarão à necessidade de métodos não-exatos (conhecidas como heurísticas), uma vez que sua formulação exata levaria a uma complexidade intratável.

A teoria da complexidade computacional tem como objetivo classificar problemas computacionais de acordo com sua dificuldade. Todo problema é tratado como sendo

um conjunto de parâmetros que definem as instâncias e um conjunto de propriedades que configuram as restrições do problema a serem satisfeitas. Assim, entre instâncias de um mesmo problema, as únicas variações estão nos conjuntos dos parâmetros [10].

Em relação ao número de computações necessárias para se obter a solução ótima, os problemas podem ser classificados em três grandes classes: problemas P, NP e Intratáveis. Em [10] temos que:

- Problemas P (*Polynomial Time*) são problemas que podem ser resolvidos em tempo polinomial. Ou seja, o esforço computacional cresce polinomialmente em função do tamanho da instância, são resolvidos no tempo $O(n^k)$ sendo k uma constante e n o tamanho da entrada. São também conhecidos como problemas tratáveis e existem algoritmos eficientes para resoluções de problemas desta classe.
- Problemas NP (*Nondeterministic Polynomial Time*) são problemas "verificáveis" em tempo polinomial. Dada uma solução para o problema, pode-se verificar em tempo polinomial se a solução satisfaz o problema de decisão, mas o esforço computacional de encontrar uma solução cresce exponencialmente em função do tamanho da instância.
- Problemas intratáveis configuram problemas onde, assim como os NP, o esforço computacional cresce exponencialmente em função do tamanho da instância, com a garantia de que a solução encontrada seja a solução ótima para o problema.

Além disso, existem duas classes denominadas NP-Difíceis e NP-Completos, sendo que os problemas que possuem uma versão de Otimização estão incluídos na classe de problemas NP-Difíceis. Os problemas NP-Difíceis e NP-Completos são problemas que não pertencem à classe P, mas possuem a dificuldade característica dos problemas da classe NP. Um problema é NP-Difícil se todos os problemas da classe NP forem polinomialmente redutíveis a ele. Ou seja, resolvendo um problema NP-Difícil em tempo polinomial, todos os problemas da classe NP também serão resolvidos em tempo polinomial. E os problemas NP-Completos englobam todos os problemas NP que também são NP-Difíceis.

Muitos problemas de otimização podem ser resolvidos usando técnicas de busca exaustiva, embora a complexidade de tempo possa ser enorme. O método *Backtracking* ou

enumeração explícita é uma maneira sistemática para percorrer todas as possíveis configurações de um espaço de busca. Essas configurações podem representar todos os arranjos possíveis de objetos (permutações) ou todas as formas possíveis de construir uma coleção deles(subconjuntos)[50]. Essa técnica consome um tempo de execução não aceitável em algumas instâncias, dependendo do tamanho da entrada. O ideal seria elaborar um algoritmo de tempo de execução aceitável, além de que a solução encontrada seja uma solução ótima para o problema. Na maioria dos casos, um algoritmo **heurístico** é utilizado para tal.

O algoritmo heurístico geralmente não cumpre uma dessas propriedades, podendo ser um algoritmo que encontra boas soluções, mas que não possui a garantia de que sempre encontrará, ou possui um processamento rápido, mas não há certeza de que será rápido para todas as instâncias do problema [49].

2.3 Considerações do capítulo

Este capítulo apresentou a fundamentação teórica necessária para o desenvolvimento e entendimento dos resultados gerados neste trabalho, principalmente em Teoria dos Grafos. Na complexidade computacional, foram relatados os principais conceitos relacionados a complexidade de algoritmos, enumeração explícita e heurísticas. O estudo destes conceitos foram de extrema importância para o desenvolvimento deste trabalho.

Capítulo 3

O Problema da Lista Coloração em Grafos e variações

Neste capítulo é abordado sobre o problema da lista coloração em grafos e algumas variações deste problema. São apresentadas as definições e conceitos principais, bem como uma tabela da revisão da literatura para lista coloração e suas variações, a μ -coloração e (γ, μ) -coloração. Este capítulo foi estruturado com base na análise realizada sobre o problema, os trabalhos e resultados encontrados.

3.1 Definições

Seja G um grafo para o qual existe um conjunto associado $L(v)$ de cores específicas para cada vértice v de G . O conjunto $L(v)$ é chamado uma **lista de cores** permitidas para v . Um exemplo é mostrado na Figura 3.1(a). O problema da lista coloração foi introduzido por Paul Erdős *et al.*, em 1979 [59] e, independentemente por Vadim Vizing em 1976 [73].

Definição 3.1 (Chartrand, 2008 [24]). *Uma **Lista Coloração** (do inglês, List Coloring) de G é uma coloração própria de G tal que $c(v) \in L(v)$ para cada $v \in V(G)$.*

Definição 3.2 (Chartrand, 2008 [24]). *Dada uma função de escolha \mathcal{L} , onde $\mathcal{L} = \{L(v) : v \in V(G)\}$. Se \mathcal{L} é o conjunto de listas de cores para os vértices de $V(G)$ e existe uma lista coloração para este conjunto \mathcal{L} de listas de cores, então G é dito ser **\mathcal{L} -lista-colorível**.*

O grafo da Figura 3.1(a) possui listas de cores associadas aos seus vértices e o mesmo grafo possui uma coloração própria de seus vértices, onde cada cor associada aos seus

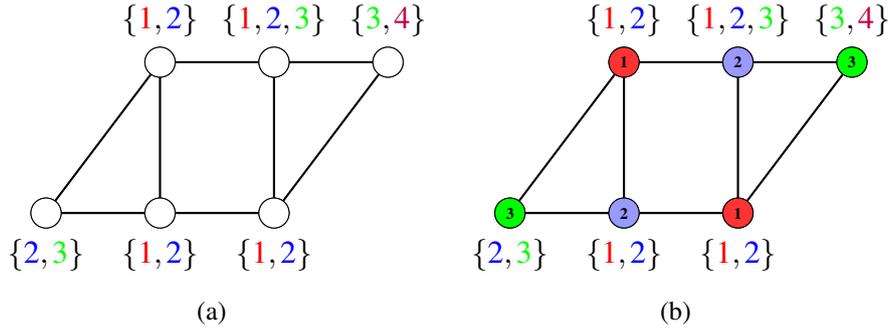


Figura 3.1: Exemplo de uma lista coloração em um grafo G . (a) Grafo com listas de cores associadas a seus vértices e (b) Grafo G com uma lista coloração.

vértices pertence à sua lista de cores, conforme é mostrada na Figura 3.1(b), logo este grafo é \mathcal{L} -lista-colorível.

Erdős *et al.* caracterizaram grafos que são d -selecionável, onde para cada vértice v de G , uma lista de cores é atribuída para cada v de tamanho $deg(v)$ (lê-se grau de v). Logo, um grafo G é d -colorível se para cada d -atribuição de listas de cores, existe uma lista coloração de G [59]. Com isso, encontraram uma generalização para o Teorema de Brooks, onde se G não é um ciclo ímpar ou um grafo completo com grau máximo Δ , então $\chi(G) = \Delta$ [4]. O mesmo acontece com lista coloração: se um grafo não é um grafo completo ou um ciclo ímpar, então G é Δ -lista-colorível [73].

3.2 (γ, μ) -coloração e μ -coloração

A μ -coloração e a (γ, μ) -coloração são algumas variações de lista coloração em grafos, introduzidas por Flávia Bonomo [19][19]. Um exemplo das duas variações de lista coloração é apresentado na Figura 3.2. As definições serão dadas a seguir:

Definição 3.3 (Bonomo, 2005 [19]). *Dado um grafo G e uma função $\mu : V(G) \rightarrow \mathbb{N}$, G é μ -colorível se existe uma coloração \mathcal{L} de G tal que $f(v) \leq \mu(v)$ para todo $v \in V(G)$.*

Definição 3.4 (Bonomo, 2006 [19]). *Dado um grafo G e uma função $\gamma, \mu : V(G) \rightarrow \mathbb{N}$ tal que $\gamma(v) \leq \mu(v)$ para todo $v \in V(G)$, G é (γ, μ) -colorível se existe uma coloração \mathcal{L} de G tal que $\gamma(v) \leq f(v) \leq \mu(v)$ para todo $v \in V(G)$.*

O esquema de generalizações em colorações por restrição, resumidos na Figura 3.3, implica que todos os problemas nessa hierarquia são polinomialmente solucionáveis em

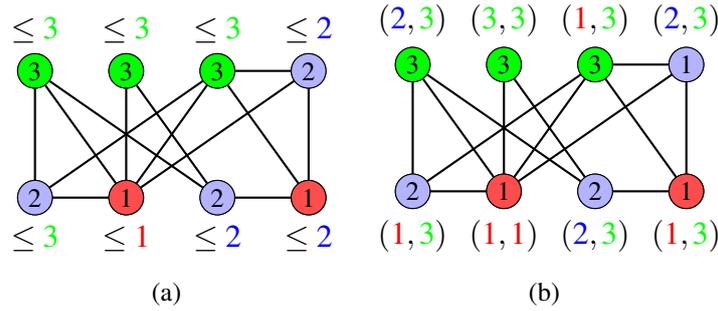


Figura 3.2: Exemplo de variações da lista coloração. (a) μ -coloração de G e (b) (γ, μ) -coloração de G .

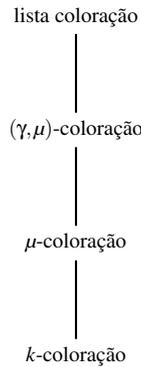


Figura 3.3: Esquema de generalizações entre problemas de coloração em grafos.

uma classe de grafos onde a lista coloração é polinomial. Por outro lado, todos os problemas são NP-Completo em uma classe de grafos onde a coloração de vértices é NP-Completo.

Existem na literatura diversos resultados com relação a lista coloração para diversas classes de grafos. O problema da lista coloração pode ser resolvido em tempo polinomial para árvores e grafos completos [38]. A Tabela 3.1 apresenta uma comparação dos resultados da literatura quanto à complexidade computacional para coloração clássica, lista coloração e suas respectivas variações, a μ -coloração e a (γ, μ) -coloração. Quase todas as classes de grafos apresentadas para lista coloração foram provadas por Jansen, em 1997 [38], com exceção da lista coloração em grafos splits, provados por Alan Bertossi [3]. Os resultados para μ -coloração e (γ, μ) -coloração foram provados por Bonomo [19][5].

Classe de Grafos	Coloração	μ-Coloração	(γ, μ)-Coloração	Lista Coloração
Geral	NP-C (Karp [35])	NP-C (Bonomo [19])	NP-C (Bonomo [5])	NP-C (Gravier [55])
Blocos	P (Bonomo [21])	P (Bonomo <i>et al.</i> [22])	P (Bonomo <i>et al.</i> [22])	P (Jansen [33])
Bipartido	P (Bonomo [5])	NP-C (Bonomo [19])	NP-C (Bonomo [5])	NP-C (Kubale [40])
Bipartido Completo	P (Bonomo [5])	P (Bonomo [5])	P (Bonomo [5])	NP-C (Jansen e Scheffler[34])
Cografos	P (Grötschel <i>et al.</i> [51])	P (Bonomo [19])	?	NP-C (Jansen e Scheffler[34])
Intervalo	P (Grötschel <i>et al.</i> [51])	NP-C (Bonomo [5])	NP-C (Bonomo [5])	NP-C (Bonomo [5])
Intervalo Unitário	P (Grötschel <i>et al.</i> [51])	NP-C (Bonomo [5])	NP-C (Bonomo [5])	NP-C (Bonomo [5])
Split	P (Golumbic [25])	NP-C (Bonomo [5])	NP-C (Bonomo [5])	NP-C (Bonomo [5])
Split Completo	P (Bonomo [5])	P (Bonomo [5])	P (Bonomo [5])	NP-C (Jansen e Scheffler[34])
Linha do K_n	P (König [39])	NP-C (Bonomo [5])	NP-C (Bonomo [5])	NP-C (Kubale [40])
Linha do $K_{n,n}$	P (König [39])	NP-C (Bonomo [5])	NP-C (Bonomo [5])	NP-C (Bonomo [5])

Tabela 3.1: Tabela de comparação da complexidade computacional para coloração, lista coloração e suas variações.

3.3 Algoritmos para lista coloração

Nesta seção, serão abordados os algoritmos em lista coloração, sendo algoritmo exato para lista coloração, um algoritmo de reconhecimento, para verificar se um grafo é \mathcal{L} -lista-colorível com as listas de cores dadas, uma heurística clássica de coloração DSATUR para lista coloração e um algoritmo para verificar se uma solução dada em lista coloração é solução válida.

O primeiro algoritmo é um método exato que verifica todas as possíveis soluções de coloração de um grafo com as listas de cores atribuídas aos seus vértices. Esse método é exato, pois testa todas as soluções. O segundo algoritmo trabalha de forma semelhante ao primeiro algoritmo, com a diferença que, encontrando uma solução válida, o algoritmo não testará todas as outras soluções. O terceiro algoritmo trata-se de uma heurística clássica da literatura para coloração de grafos, o DSATUR, que foi proposto por Daniel Brélez em 1979 [7], que trabalha com o grau de saturação dos vértices de um grafo. O último algoritmo abordado trata-se de um algoritmo para validar uma solução em lista coloração. Dado uma instância, verificar se a mesma é uma solução válida para lista coloração. Nas próximas seções, os algoritmos serão detalhados.

3.3.1 Algoritmo para gerar todas as soluções para lista coloração

O primeiro algoritmo trata-se de um método exato que verifica todas as soluções possíveis. Esse método encontrará sempre uma solução, se ela existir. Por outro lado, esse método tem se tornado incapaz de fornecer soluções em tempo computacional aceitável.

O algoritmo exato abordado é para lista coloração. O algoritmo testa todas as soluções em lista coloração e armazena as soluções que são válidas. O método, que é recursivo, inicia com a condição de parada *falso*. Esse controle é feito pela variável *existe*, na linha 1. Como no início da execução o grafo ainda não está colorido, o algoritmo segue para a linha 8 para o conjunto de listas L , e assim, na linha 10, o vértice é colorido com as cores de sua respectiva lista de cores e o algoritmo faz uma chamada recursiva para colorir o vértice $i + 1$, até que todos os vértices recebam a primeira cor da sua lista de cores.

Algoritmo 1: Lista_Coloracao_Exato(G, L, i)

```
1 se todas as soluções não foram testadas então
2   se  $L = \emptyset$  então
3     se ExisteColoracao( $G$ ) então
4       Armazena solução válida;
5       cont++
6     fim se
7   senão
8      $l \leftarrow$  remove o primeiro da lista  $L$ 
9     para cada  $cor \in l$  faça
10       $G.v_i.cor \leftarrow$  cor
11      Lista_Coloracao_Exato( $G, L, i + 1$ )
12    fim para cada
13  fim se
14 fim se
```

Se todos os vértices estão coloridos, mas não existe uma coloração própria para o grafo, o algoritmo segue para a próxima cor da lista de cores do último vértice visitado. Se ainda assim não existir coloração própria e não houver mais cores disponíveis na lista de cores à serem testados, o algoritmo volta um nível na árvore de execução para testar a próxima cor disponível desse vértice, e assim por diante. A variável *cont*, na Linha 5 conta a quantidade de soluções encontradas e na Linha 4, a solução encontrada é armazenada. Se o algoritmo encontrar uma coloração válida, a variável *existe* recebe *sim* e ao retornar para última parada, o algoritmo conta uma solução válida e continua a execução do código até que não haja mais cores na lista à serem testadas.

Um exemplo de execução do algoritmo é apresentado através do grafo ciclo da Figura 3.4. Todas as possíveis soluções de coloração são visitados nesse algoritmo. Essas soluções são visíveis pela árvore de execução. As linhas tracejadas indica o nível da árvore, que é determinado pelo número de vértices, ou seja, o v_1 , que possui lista de cor $\{1, 2\}$, só pode receber essas duas cores, logo no nível desse vértice somente essas cores estarão disponíveis.

A árvore de execução do Algoritmo 1 é apresentado na Figura 3.4. O algoritmo per-

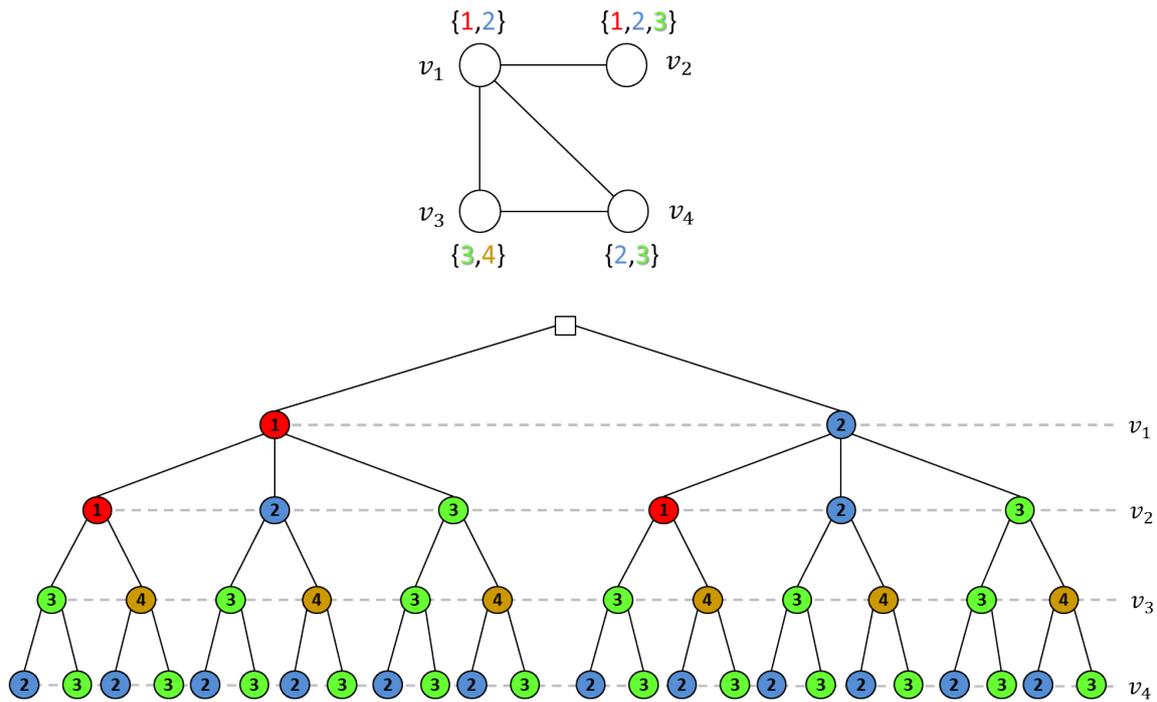


Figura 3.4: Exemplo de um grafo com suas listas de cores e a árvore gerada pela execução do Algoritmo 2 para o grafo.

corre todas as soluções de possíveis colorações para o grafo. Ao final de cada nível, o **X** indica que a coloração para o grafo do exemplo é inválida. Já o **✓** indica que a coloração é válida. Ao final da execução, o algoritmo contabiliza um total de 8 (oito) soluções de coloração para o grafo do exemplo, com as listas de cores dadas.

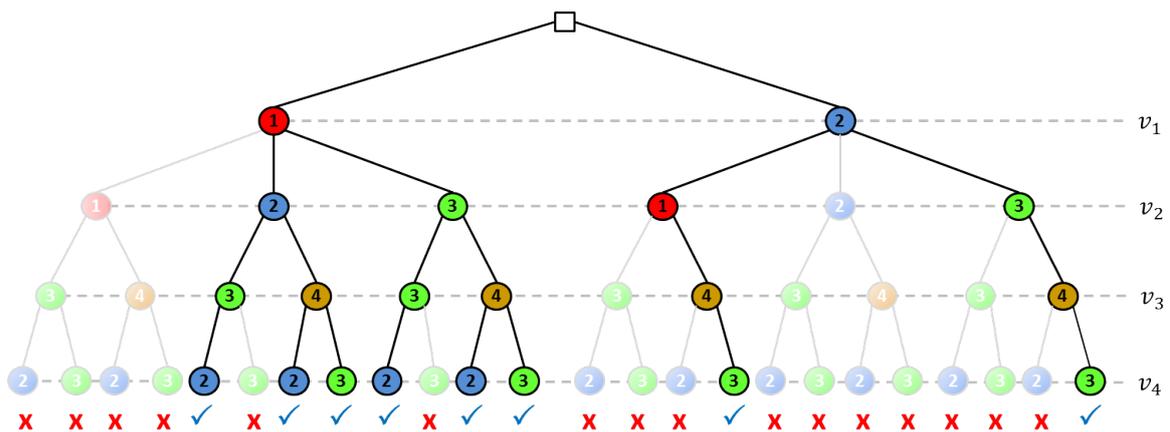


Figura 3.5: Exemplo da árvore gerada pela execução do Algoritmo 2 para o grafo da Figura 3.4 com as soluções encontradas em destaque.

O algoritmo testa todas as possíveis soluções de coloração para as listas de cores atribuídas para um grafo. A Figura 3.5 apresenta as soluções que são inválidas que foram descartadas, cujo o sombreamento está mais claro e as soluções válidas que foram consi-

deradas e contadas na Linha 5 do Algoritmo 1. A Figura 3.6 apresenta todas as soluções encontradas para o grafo do exemplo.

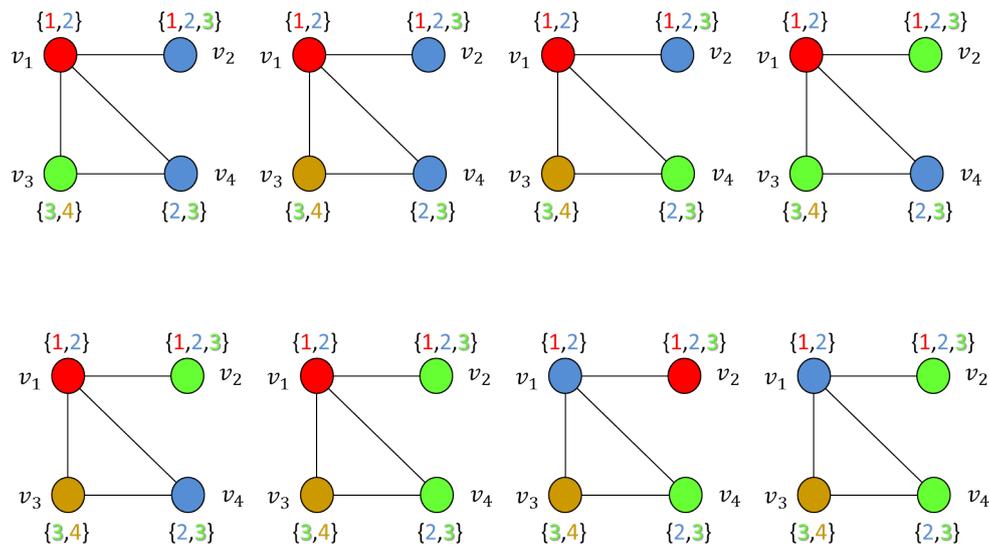


Figura 3.6: Exemplo das soluções encontradas.

Considerando que cada vértice possui uma lista de cor, denotado por $|L(v)|$, o algoritmo gera no total $S = \prod_{i=1}^n |L(v_i)|$, onde S é a quantidade de soluções geradas. Portanto, para o exemplo da Figura 3.4 foram geradas um total de 24 soluções de coloração para as listas de cores $\{1, 2\}$, $\{1, 2, 3\}$, $\{2, 3\}$ e $\{3, 4\}$.

O Algoritmo 1 possui complexidade computacional $O(n^p)$, onde n é o número de vértices e p é dado pelo tamanho do maior conjunto de listas de cores. Considerando que o tamanho da maior lista de cores pode ser n , a complexidade computacional será $O(n^n)$.

3.3.2 Algoritmo para determinar uma Lista Coloração de um grafo

O algoritmo apresentado nesta seção verifica todas as soluções possíveis, porém, quando o algoritmo encontra uma coloração válida, o restante das possíveis soluções não precisam ser mais exploradas. Ou seja, algoritmo verifica se existe uma coloração própria do grafo com as listas de cores associadas aos vértices. Quando o algoritmo encontra uma solução válida, ou seja, encontra uma coloração própria para o grafo, o restante da árvore de enumeração de possíveis soluções não precisa ser visitado. O pseudocódigo está descrito no Algoritmo 2.

O algoritmo, que é recursivo, inicia com a condição de parada *falso*. Esse controle é feito pela variável *existe*, na linha 1. Como no início da execução o grafo ainda não está

colorido, o algoritmo segue para a linha 7 para o conjunto de listas L , e assim, na linha 9, o vértice é colorido com as cores de sua respectiva lista de cores e o algoritmo chama recursivamente o algoritmo para colorir o vértice $i + 1$, até que todos os vértices estejam coloridos.

Algoritmo 2: $Existe_Lista_Coloracao(G, L, existe, i)$

```

1 se  $!(existe)$  então
2   se  $L = \emptyset$  então
3     se  $ExisteColoracao(G)$  então
4       existe  $\leftarrow$  SIM;
5     fim se
6   senão
7      $l \leftarrow$  remove o primeiro da lista  $L$ ;
8     para cada  $cor \in l$  faça
9        $G.v_i.cor \leftarrow cor$ ;
10       $Existe\_Lista\_Coloracao(G, L, existe, i + 1)$ ;
11      se  $(existe)$  então
12        Interrompa;
13      fim se
14    fim para cada
15  fim se
16 fim se

```

Logo, se existe uma coloração para o grafo com a lista de cores dada, o algoritmo termina sua execução (linha 12). O restante das soluções não serão exploradas.

Se todos os vértices estão coloridos, mas não existe uma coloração própria para o grafo, o algoritmo segue para a próxima cor da lista de cores do último vértice visitado. Se ainda assim não existir coloração própria e não houver mais cores disponíveis na lista de cores à serem testados, o algoritmo volta um nível na árvore de execução (veja Figura 3.8) para testar a próxima cor disponível desse vértice, e assim por diante.

A árvore de execução gerada pelo Algoritmo 2 na Figura 3.8. Primeiramente, o algoritmo começa atribuindo a primeira cor da lista de cores do vértice v_1 . O algoritmo segue atribuindo aos vértices suas respectivas cores de suas listas até que não haja mais vértices

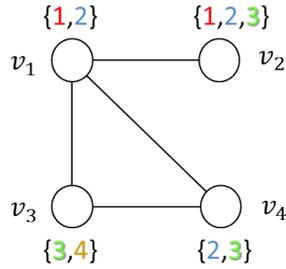


Figura 3.7: Exemplo de um grafo com listas de cores para demonstrar a execução do Algoritmo 2

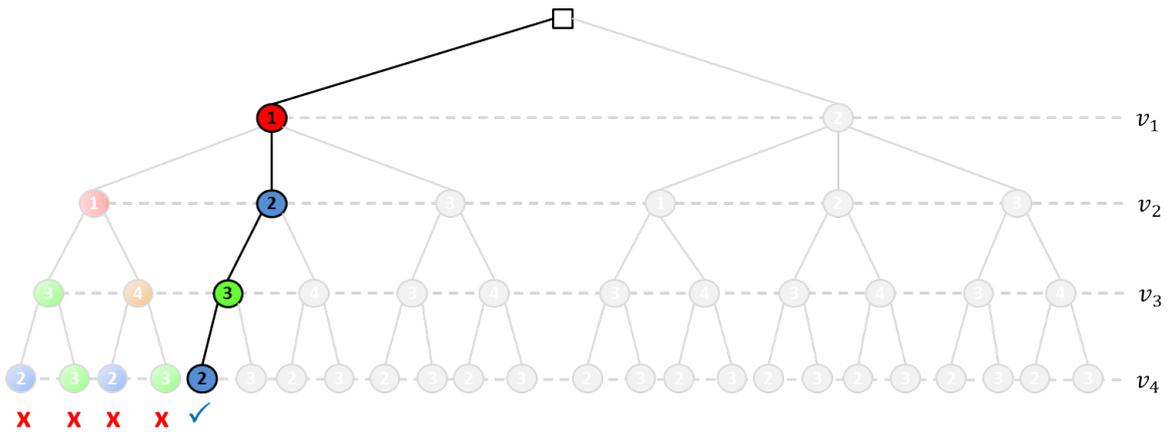


Figura 3.8: Exemplo da árvore gerada pela execução do Algoritmo 2 para o grafo da Figura 3.7. O algoritmo encontra uma solução válida e não visita mais o restante das soluções.

à serem coloridos, no exemplo da Figura 3.8, até o vértice v_4 . Quando isso acontece, a condição de parada da linha 2 do Algoritmo 2 é satisfeita. No entanto, a coloração não é válida e o algoritmo *volta* um nível na árvore, seguindo assim para a próxima cor do vértice v_3 e seguindo novamente até o vértice v_4 para testar as cores de sua lista de cores novamente. Esse processo é repetido até que se encontre uma coloração válida para o conjunto de listas de cores associadas ao grafo.

No Algoritmo 1, o mesmo executa todas as soluções possíveis. Nesse caso, o Algoritmo 2 apenas encontra uma solução válida. Porém, o cálculo da quantidade de soluções de listas de cores permanece, pois o algoritmo pode encontrar uma solução válida no final da árvore de execução, portanto o algoritmo gera no total $S = \prod_{i=1}^n |L(v_i)|$, onde S é a quantidade de soluções geradas.

A complexidade computacional do Algoritmo 2 é semelhante ao Algoritmo 1, $O(n^p)$, onde n é o número de vértices e p é dado pelo tamanho do maior conjunto de listas de cores. Considerando que o tamanho da maior lista de cores pode ser n , a complexidade

computacional será $O(n^n)$.

No Capítulo 4, Seção 4.4, será apresentado um algoritmo de enumeração explícita para verificar se um grafo é k -selecionável, que utiliza o Algoritmo 2.

3.3.3 Heurística gulosa DSATUR para lista coloração

O terceiro algoritmo abordado é uma clássica heurística para coloração de grafos, o DSATUR (sigla para *Degree of Saturation*), que é um algoritmo heurístico que utiliza uma abordagem gulosa, levando em conta o grau de saturação dos vértices. Foi proposto por Daniel Brélaz em 1979, onde o autor também apresentou uma prova de que esse algoritmo é exato para grafos bipartidos [7]. A sequência de passos do DSATUR será descrito a seguir:

1. Inicialmente, o algoritmo ordena os vértices por ordem decrescente de grau. A ideia do algoritmo é colorir primeiramente os vértices de maior grau;
2. O vértice de maior grau é inicialmente colorido com a cor 1. Neste momento, o grau de saturação dos vértices vizinhos ao vértice de maior é acrescido de um;
3. Na próxima iteração, seleciona-se o vértice de maior grau de saturação, se existir algum vértice com o mesmo grau de saturação, escolhe-se o vértice de maior grau. Persistindo o empate, escolha é feita pelo vértice de menor índice. Por fim, o vértice escolhido será colorido com a menor cor disponível, tal que essa cor não conflite com seus vértices adjacentes;
4. Repete-se a iteração até que todos os vértices estejam coloridos.

O pseudocódigo do DSATUR está descrito no Algoritmo 3. O algoritmo recebe como entrada um grafo G e inicia ordenando os vértices por ordem decrescente de grau, na linha 2. Logo após, na linha 2, ao vértice de maior grau é atribuída a cor 1. Esse processo pode ser visto também na Figura 3.9(b), onde o vértice a é escolhido como o vértice de maior grau e recebe a cor 1 (vermelha). Nas linhas 6-8, o algoritmo procura o vértice de maior grau de saturação. Se existir outro vértice com grau de saturação maior, o algoritmo escolhe o vértice de maior grau. Com o vértice escolhido, o mesmo é finalmente colorido. O processo se repete até que o grafo seja totalmente colorido. A seguir, um exemplo de execução do algoritmo DSATUR em um grafo dado:

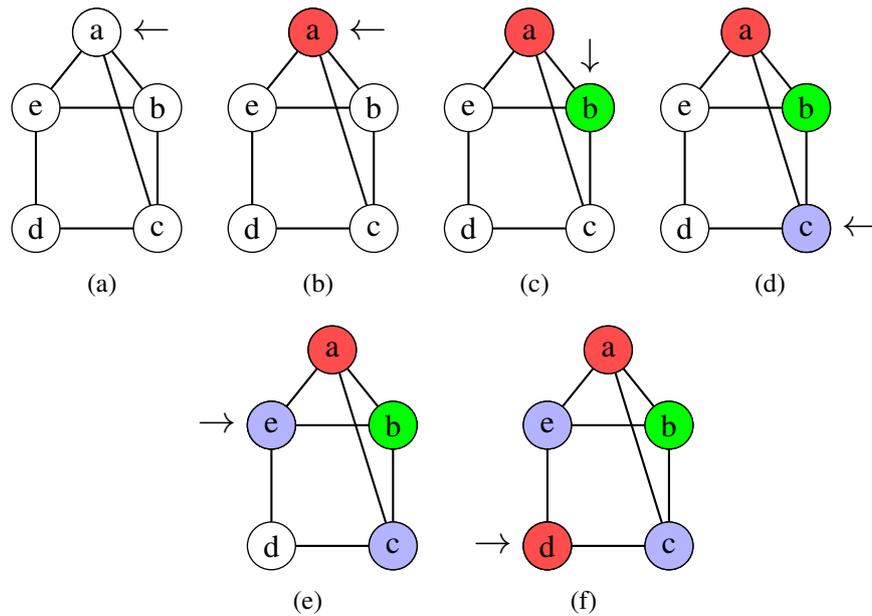


Figura 3.9: Exemplo de execução do Algoritmo DSATUR.

Algoritmo 3: DSATUR($G = (V, E)$)

```

1 início
2   Ordenar os vértices  $V(G)$  em ordem decrescente de grau;
3   Vértice de maior grau recebe a cor 1;
4   enquanto  $\exists$  vértice descolorido faça
5      $v \leftarrow \text{grau\_max\_saturao}(G)$ ;
6     se  $\exists$  mais de um vértice possui grau de saturação igual a  $v$  então
7       Escolhe o vértice de maior grau;
8     fim se
9     Colore o vértice com a menor cor disponível
10  fim enquanto
11 fim

```

O algoritmo DSATUR foi utilizado para um grafo com coloração por listas, mostrado no Algoritmo 4. No algoritmo, em vez da utilização da menor cor disponível no momento de colorir, serão utilizadas as cores disponíveis na lista de cores de cada vértice $v \in V(G)$.

No Algoritmo 4, o processo de coloração é semelhante ao DSATUR original, porém, as cores utilizadas pertencem à lista de cores do vértice. Esse processo pode ser visto na

linha 3, onde o vértice de maior grau recebe uma cor de sua lista de cores. O mesmo ocorre no momento de colorir o vértice com maior grau de saturação, a cor é escolhida da lista de cores desse vértice, sem conflitar com o seu vértice adjacente. As setas indicam a iteração no algoritmo. Na Figura 3.9(a), o primeiro passo é escolher o vértice de maior grau, no exemplo, o vértice a . Na Figura 3.9(b), o vértice de maior grau é colorido com 1 (vermelho). Já na Figura 3.9(c) o próximo vértice a ser escolhido é o de maior grau de saturação, porém, ocorre um empate, o algoritmo escolhe o de maior grau, no caso, vértice b , onde b é colorido com 2 (verde). Na Figura 3.9(d), o fato se repete, o vértice c é escolhido e colorido com 3 (azul). Na Figura 3.9(e), o vértice de maior grau de saturação é escolhido e colorido com 3 e por fim, na Figura 3.9(f) o último vértice é colorido com 1.

O algoritmo, por sua característica gulosa, não funciona em alguns casos de grafos, veja Figura 3.10. Observe que na Figura 3.10(b), quando o algoritmo tenta colorir o vértice e , as cores disponíveis em sua lista de cores já foram utilizadas nos vértices a e b , não sendo possível colorir o grafo. No exemplo da Figura 3.10(a), é possível obter uma coloração própria para o grafo com as listas de cores disponíveis.

Algoritmo 4: L-DSATUR($G = (V, E)$)

```

1 início
2   Ordenar os vértices  $V(G)$  em ordem decrescente de grau;
3   Vértice  $v$  de maior grau recebe a cor  $c \in L(v)$ ;
4   enquanto  $\exists$  vértice descolorido faça
5      $v \leftarrow \text{grau\_max\_saturao}(G)$ ;
6     se  $\exists$  mais de um vértice possui grau de saturação igual então
7       Escolhe o vértice de maior grau;
8     fim se
9     Colore o vértice  $v_i$  com uma cor de sua lista de cores  $L(v)$ , tal que essa cor
       não conflita com seu vértice adjacente
10  fim enquanto
11 fim

```

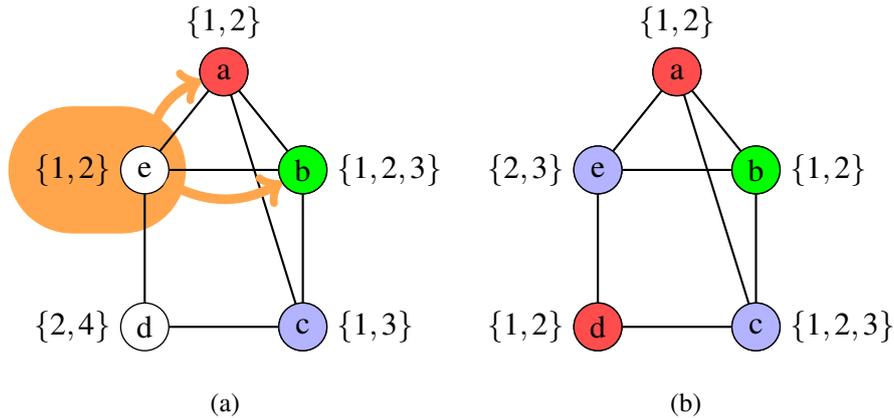


Figura 3.10: Exemplo de execução do Algoritmo L-DSATUR. Em (a), um exemplo de um grafo com restrição de cores onde o L-DSATUR não funciona. Em (b), um exemplo onde L-DSATUR funciona.

A complexidade computacional do Algoritmo DSATUR é de $O(n^2)$ e do Algoritmo DSATUR para lista coloração também é $O(n^2)$ e é uma heurística para encontrar uma rápida coloração de um grafo. Na próxima seção será apresentado um algoritmo de reconhecimento para validar uma solução em lista coloração.

3.3.4 Algoritmo para reconhecimento de uma solução em lista coloração

O quarto algoritmo abordado é para verificar se dada uma instância de um grafo colorido com as cores de suas listas de cores, verificar se é solução válida.

O algoritmo inicia verificando se as cores atribuídas aos vértices de um grafo não ferem a restrição de vértices adjacentes que devem receber cores diferentes. Esse passo pode ser verificado na Linha 2.

O algoritmo verifica na linha 9 se o a cor do vértice v pertence à sua lista de cores $L(v)$. Se for verdade, o algoritmo segue e testa para outros vértices do grafo. Senão, não há necessidade de seguir adiante, pois se a cor do vértices não pertence à sua lista de cores correspondente, então não é solução válida.

O algoritmo percorre todo o grafo, ou seja, verifica para cada vértice se a cor atribuída é diferente à de seus vértices adjacentes e se sua cor pertence à sua lista de cores. Sua complexidade computacional é de $O(\max\{|V|, L_{max}\})$, onde L_{max} é o tamanho da maior lista de cores do grafo.

Algoritmo 5: Validar_Lista_Coloracao($G = (V, E)$)

```
1 para cada  $v \in V$  faça
2   para cada  $u$  adjacente a  $v$  faça
3     se  $cor(u) = cor(v)$  então
4       Solução inválida
5     fim se
6   fim para cada
7 fim para cada
8 para cada  $v \in V$  faça
9   se a  $cor(v) \in L(v)$  então
10    Testar os vértices restantes
11  senão
12    Solução inválida
13  fim se
14 fim para cada
```

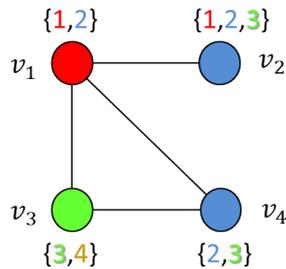


Figura 3.11: Exemplo de uma solução para demonstrar a execução do Algoritmo 2.

Um exemplo pode ser verificado na solução da Figura 3.11. O algoritmo irá verificar se a mesma é solução em lista coloração. O algoritmo irá verificar para cada vértice se a cor atribuída não viola a regra de coloração própria e se cada uma dessas cores pertencem à suas respectivas lista de cores. No exemplo, é possível verificar que a cor atribuída ao vértice também está contida na sua respectiva lista de cores. Portanto a mesma é uma solução válida para lista coloração em grafos.

3.4 Considerações do capítulo

Este capítulo apresentou as principais definições sobre o problema da lista coloração em grafos, suas principais variações, a μ -coloração e a (γ, μ) -coloração. Foi apresentado uma

tabela comparando os principais resultados na literatura com essas colorações. Foi apresentado também quatro abordagens algorítmicas em lista coloração de forma detalhada e a complexidade computacional de cada uma delas.

Capítulo 4

A Propriedade da Seleccionabilidade em Grafos

Neste capítulo é abordada a seleccionabilidade em grafos, muito diretamente relacionada com a lista coloração em grafos, apresentada no Capítulo 3. Serão apresentadas as definições, conceitos principais e provas das principais propriedades em seleccionabilidade. Este capítulo foi estruturado com base na análise realizada sobre o problema, e os trabalhos e resultados encontrados.

4.1 Definições

A lista coloração está fortemente ligada com **seleccionabilidade em grafos**, que também foi introduzido por Paul Erdős, Arthur Rubin e Herbert Taylor, em 1979 [59].

Definição 4.1 (Chartrand, 2008 [24]). *Um grafo G é k -seleccionável (ou k -choosable) se G é \mathcal{L} -lista-colorível (ou \mathcal{L} -seleccionável) para toda coleção \mathcal{L} de listas $L(v)$ para os vértices de G tal que $|L(v)| \leq k$ para todo $v \in V(G)$.*

Definição 4.2 (Chartrand, 2008 [24]). *A lista número cromático (do inglês, list chromatic number), $\chi_\ell(G)$ de G é o mínimo inteiro positivo k tal que G é k -seleccionável.*

Os autores apresentaram uma caracterização de grafos que são 2-seleccionável. Em 2008, Chartran [24] mostrou que grafos árvores são 2-seleccionável. Um grafo árvore pertence à uma subclasse de grafos bipartidos, portanto, pensa-se naturalmente que um grafo bipartido $G[X, Y]$ seria 2-seleccionável, pois, na partição X , têm-se os vértices coloridos

com a cor a e na partição Y , os vértices coloridos com a cor b . Será demonstrado um tipo de grafo bipartido onde esse processo não funciona.

Considere o grafo bipartido completo $K_{3,3}$, mostrado na Figura 4.1. Dadas as listas de cores associadas à seus vértices, colore-se v_1 com a cor 1. Logo, v_4 é colorido com a cor 2, v_5 é colorido com 3 e v_6 com a cor 2. No momento de colorir o vértice v_3 , observa-se que as cores associadas à sua lista de cores, $L(v_3) = \{2,3\}$, já foram utilizadas em v_4 e v_5 . Logo, é necessário uma cor extra (que não está em sua lista de cores) para colori-lo. Portanto, $K_{3,3}$ não é 2-selecionável e sim 3-selecionável.

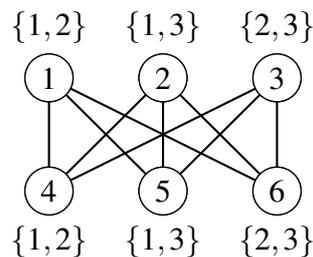


Figura 4.1: Exemplo de um grafo bipartido $K_{3,3}$ que não é 2-selecionável.

O exemplo do grafo $G = K_{3,3}$ mostra que $\chi_\ell(G) > \chi(G)$. Uma característica interessante em grafos bipartidos completos $K_{r,r}$, é que se $r = \binom{2k-1}{k}$, então $K_{r,r}$ não é k -selecionável [59].

Erdős *et al.* caracterizaram grafos que são 2-selecionáveis. Os mesmos apresentaram o θ -grafo. Um θ -grafo consiste de dois vértices u e v que conectam três caminhos internos disjuntos. O grafo $\theta_{i,j,k}$ é o θ -grafo cujo os três caminhos internos disjuntos possuem tamanhos i, j e k [59]. Um exemplo deste grafo é mostrado na Figura 4.2.

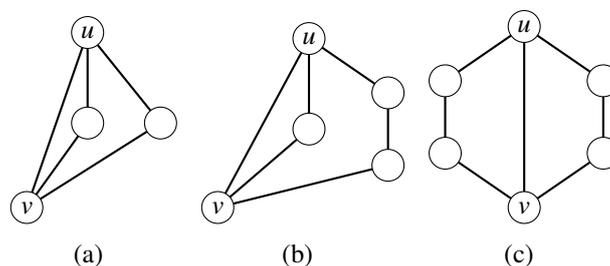


Figura 4.2: Exemplos de θ -Grafos: (a) $\theta_{1,2,2}$, (b) $\theta_{1,2,3}$ e (c) $\theta_{1,3,3}$

Teorema 4.1 (Erdős *et al.*[59]). Para $k \geq 1$, o grafo $\theta_{2,2,2k}$ é 2-selecionável.

Prova. Considere os seguintes casos: Caso 1. Todas as listas de cores são **iguais** e Caso 2. Nem todas as listas de cores são iguais.

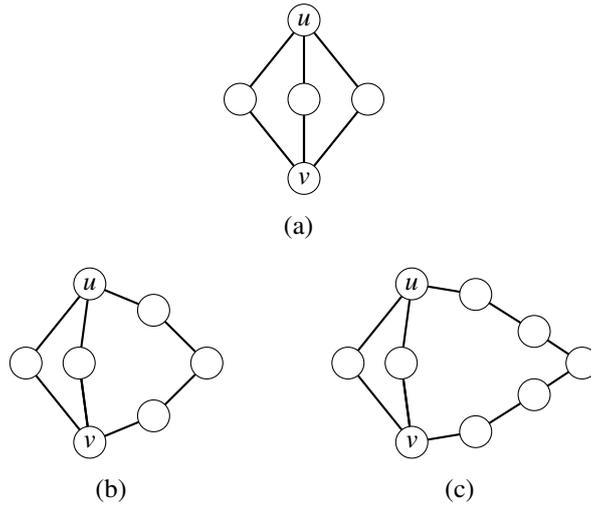


Figura 4.3: Exemplos de Grafos $\theta_{2,2,2k}$: (a) $\theta_{2,2,2}$, (b) $\theta_{2,2,4}$ e (c) $\theta_{2,2,6}$

Caso 1. Todas as listas de cores são iguais, digamos $\{a, b\}$. Observe a rotulagem deste grafo na Figura 4.4. Seja um vértice v_i . Se i for ímpar, esse vértice recebe a cor a . Se i for par, então v_i recebe a cor b . Observe que os vértices v_1 e v_{2k+1} , que são ímpares, recebem a cor a , portanto a cor dos vértices x e y será a cor b . Logo, uma coloração própria.

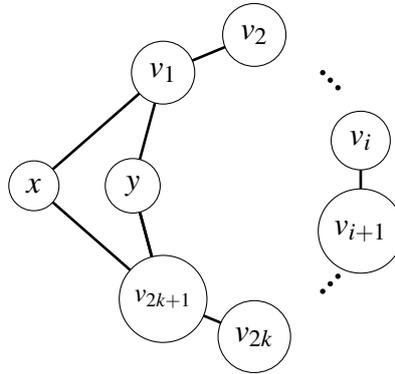


Figura 4.4: Exemplo de um Grafo $\theta_{2,2,2k}$ rotulado para a prova.

Caso 2. Nem todas as listas de cores são iguais. Utilizar as mesmas rotulagens da Figura 4.4, encontre dois vértices adjacentes v_i e v_{i+1} tal que $L(v_i) \neq L(v_{i+1})$. Colorir o grafo da seguinte forma: Colorir v_i com uma cor de sua lista que não está na lista de v_{i+1} . Continuar essa coloração até colorir v_1 com a cor c_v . Observe a lista de cores de v_{2k+1} . Seja $L(v_{2k+1}) = \{c_1, c_2\}$. Se $L(x) \neq \{c_v, c_1\}$ e $L(y) \neq \{c_v, c_2\}$ então existe uma cor $c_{2k+1} \in L(v_{2k+1})$ tal que $L(x) - \{c_v, c_{2k+1}\} \neq \emptyset$ e $L(y) - \{c_v, c_{2k+1}\} \neq \emptyset$. Escolher as cores para x e y e continue colorindo até v_{i+1} . Então temos uma coloração própria.

Mas se $L(x) = \{c_v, c_1\}$ e $L(y) = \{c_v, c_2\}$ então voltamos ao ponto inicial v_i e v_{i+1} . Escolher a cor para $L(v_{i+1})$ que não está em $L(v_i)$ agora faça a coloração em sentido

horário até colorir v_{2k+1} com uma cor $c_v \neq L(v_1)$. Escolher a cor c_v para os vértices x e y . Existe uma cor disponível em v_1 . Continuar colorindo o grafo e verifica-se que existe uma coloração própria. \square

Uma vez que o ciclo par C_{2k+2} é um subgrafo de $\theta_{2,2,2k}$, temos que todo ciclo par é 2-seleccionável [59]. Todo grafo é 2-seleccionável se possuir em seu núcleo um $\theta_{2,2,2k}$. Um núcleo de um grafo é a sucessiva poda de seus vértices de grau 1 (folhas). O subgrafo restante é o núcleo do grafo. É possível sempre colorir esses nós (folhas) com uma cor da lista de cores de tamanho 2, e a outra cor restante será atribuída para seu vértice adjacente [59].

Teorema 4.2 (Erdős *et al.* [59]). *Um grafo é 2-seleccionável se, e somente se, o núcleo do grafo pertence a $T = \{K_1, C_{2k+2}, \theta_{2,2,2k}\}$.*

Prova. Seja G o núcleo de um grafo conectado. A ideia da prova é mostrar, por exaustão, todas as possibilidades, que ou G está em T , ou G contém um subgrafo que pertence a algum dos cinco tipos de grafos seguintes:

- Um ciclo ímpar, mostrado na Figura 4.5(a);
- Dois vértices de ciclos pares disjuntos conectados por um caminho, mostrado na Figura 4.5(b);
- Dois ciclos pares que possuem exatamente um vértice em comum, mostrado na Figura 4.5(c);
- Grafos $\theta_{a,b,c}$, onde $a \neq 2$ e $b \neq 2$, mostrado na Figura 4.5(d);
- Grafos $\theta_{2,2,2,2k}$, mostrado na Figura 4.5(e).

Assuma que G não pertence a T . Se G contém um ciclo ímpar, o mesmo é um dos cinco tipos citados acima, portanto não é um $K_1, C_{2k+2}, \theta_{2,2,2k}$ e a prova pode ser finalizada.

Então, suponha que G é bipartido. Seja C_1 o menor ciclo. Se existe um ciclo C_2 , haverá um vértice em comum com o ciclo C_1 , temos o grafo tipo 2 e o grafo tipo 3 e então a prova é finalizada.

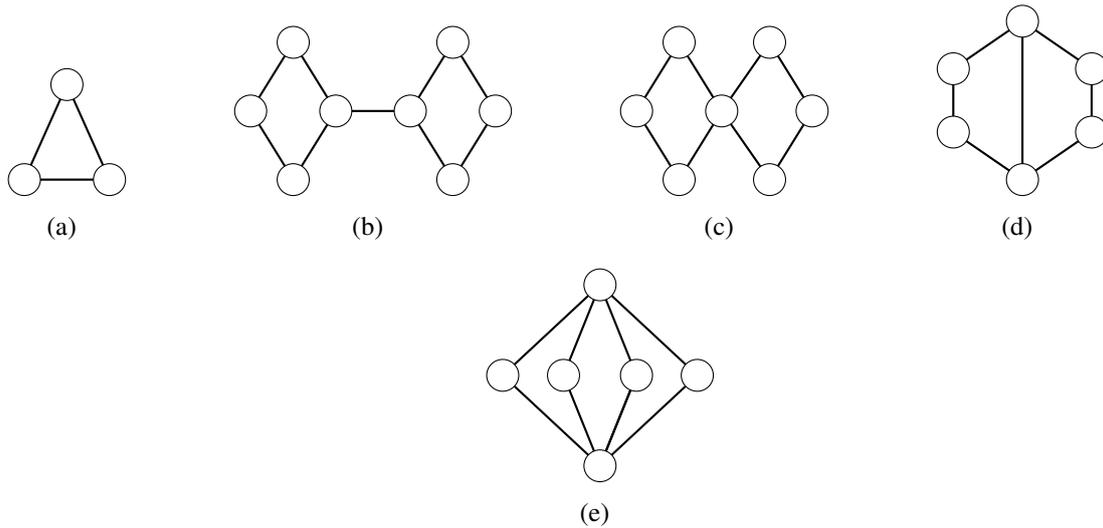


Figura 4.5: Exemplo de grafos para prova: (a) Grafo Tipo 1, (b) Grafo Tipo 2, (c) Grafo Tipo 3, (d) Grafo Tipo 4 e (e) Grafo Tipo 5.

Seja P_1 o menor caminho com arestas disjuntas de C_1 e que conecta dois vértices disjuntos de C_1 . Se $C_1 \cup P_1$ não está em T , então temos o grafo tipo 4 e então a prova é finalizada.

Agora, suponha que $C_1 \cup P_1$ está em T . Se está em T , então ele é do tipo $\theta_{2,2,2k}$ e C_1 é um 4-ciclo. Seja P_2 o menor caminho tal que suas arestas são disjuntas de $C_1 \cup P_1$, mas conectam dois vértices distintos de $C_1 \cup P_1$. Então há seis casos à serem considerados:

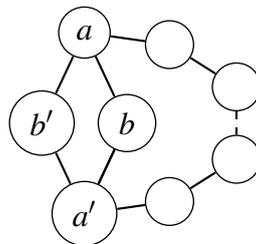


Figura 4.6: Grafo $C_1 \cup P_1$ para a prova.

Caso 1. Se as extremidades de P_2 são dois vértices do interior de P_1 então existe um ciclo disjunto de C_1 , logo haverá um grafo do tipo 2 novamente (Figura 4.7(a)).

Caso 2. Se as extremidades de P_2 são: o vértice a de C_1 e um vértice do interior de P_1 , então existe um ciclo com exatamente um vértice em comum com C_1 , e haverá um grafo do tipo 3 (Figura 4.7(b)).

Caso 3. Se as extremidades de P_2 são b e um vértice do interior de P_1 , então existe um caminho de a para b sem alguma aresta de C_1 , existe um grafo do tipo 4 (Figura 4.8(a)).

Caso 4. Se as extremidades de P_2 são a e b , então o caso 4 ocorre novamente (Figura

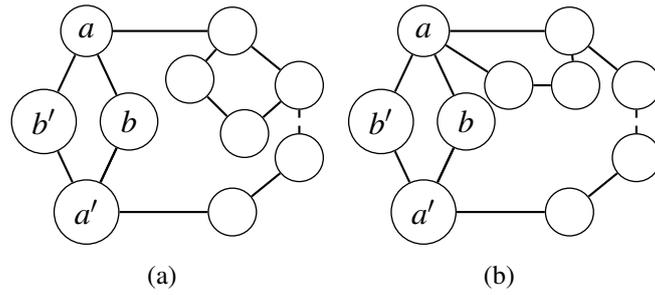


Figura 4.7: Exemplo de grafos para a prova. O grafo (a) para o caso 1 e (b) para o caso 2.

4.8(b)).

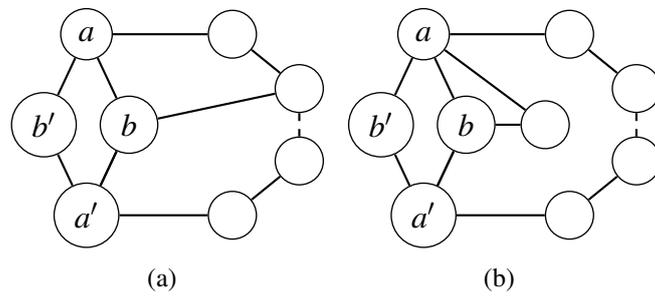


Figura 4.8: Exemplo de grafos para a prova. O grafo (a) para o caso 3 e (b) para o caso 4.

Caso 5. Se as extremidades de P_2 são a e a' , e P_1 é de tamanho 2, existe um grafo do tipo 5. Se P_1 é de tamanho maior que 2, existe um grafo do tipo 4.

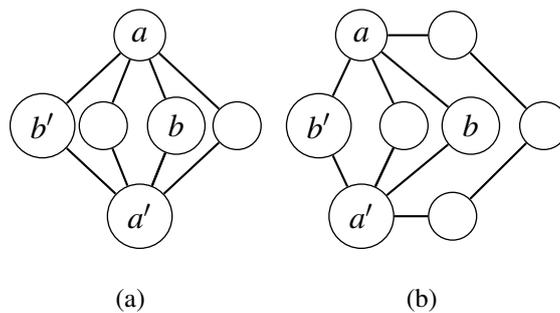


Figura 4.9: Exemplo de grafos para a prova. O grafo (a) para o caso 5, onde o tamanho de P_1 é 2 e em (b), o tamanho de P_1 é maior que 2.

Caso 6. Se as extremidades de P_2 são b e b' , removendo qualquer aresta de C_1 , encontramos um grafo θ e existe um grafo do tipo 4.

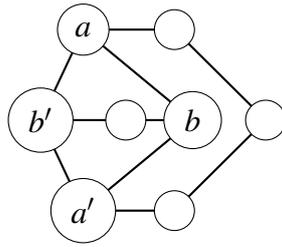


Figura 4.10: Grafo onde as extremidades de P_2 são b e b' .

Na Figura 4.11 temos o exemplo dos cinco tipos de grafos. Observe que os mesmos possuem listas de cores de tamanho dois, onde não é possível obter uma \mathcal{L} -lista-coloração.

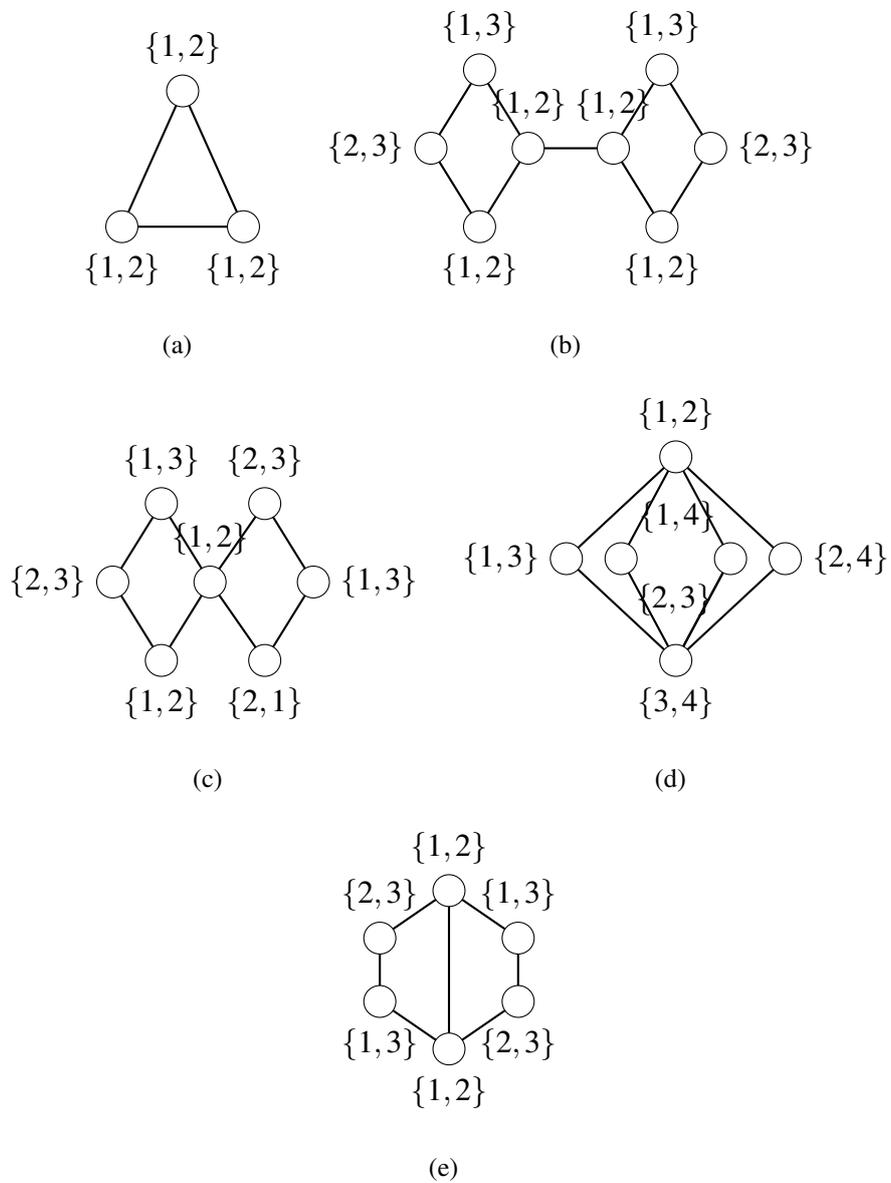


Figura 4.11: Exemplo de listas de cores para os cinco tipos de grafos que não são 2-selecionáveis.

Então, se G não está em T , G é um dos cinco tipos de grafos. A Figura 4.11, apresenta os cinco tipos de grafos não são 2 -selecionável. E assim a prova é finalizada. \square

4.2 Técnicas de provas em selecionabilidade

Em 2001, o autor Douglas Woodall [80] apresentou as principais técnicas de provas utilizadas nos resultados existentes na literatura em selecionabilidade de grafos. Essas técnicas podem ser divididas em técnicas que envolvem digrafos e técnicas que não envolvem digrafos.

4.2.1 Técnica de degeneração em grafos

Um grafo é k -degenerado se pode ser reduzido a um K_1 por repetidas remoções de seus vértices de grau no máximo k [4]. Um exemplo de grafos degenerados são grafos árvores, que são 1-degenerados. Grafos planares são grafos 5-degenerados, pois possuem um vértice de até no máximo grau 5, assim como grafos periplanares, que são 2-degenerados [45].

Um grafo que é k -degenerado é $k + 1$ -selecionável [79]. Grafos planares são 6-selecionável com base na afirmação anterior: grafos planares são 5-degenerados e portanto, 6-selecionável [32][80]. Em 1994, Kratochvíl e Tuza [32] mostraram que grafos triângulo-livre são 3-degenerados, portanto são 4-selecionável. Grafos bipartidos são grafos triângulo-livre, portanto, pensa-se naturalmente que o argumento anterior vale para este tipo de grafo, porém, Alon e Tarsi [57] demonstraram (através da técnica de prova Alon-Tarsi) que grafos bipartidos planares são 3-selecionável.

Classe de Grafo	k -degeneração	k -selecionabilidade	Referências
Planares	5-degenerado	6-selecionável	[32], [80]
Periplanares	2-degenerado	3-selecionável	[45]
Livres de triângulo	3-degenerado	4-selecionável	[32]
Bipartido Planar	3-degenerado	3-selecionável	[57]
Arco-Circulares com cintura 5 ou menos	2-degenerado	3-selecionável	[1]
Planares livre de C_5	3-degenerado	4-selecionável	[79]
Planares livre de C_7	3-degenerado	4-selecionável	[26]
Planares de cintura no máximo 5	3-degenerado	3-selecionável	[70]

Tabela 4.1: Resultados de selecionabilidade em grafos que utilizam como prova a técnica de degeneração em grafos.

4.2.2 Técnica do sistema dos representantes distintos

É uma técnica que utiliza o Sistema dos Representantes Distintos (SRD), introduzido por Marshal Hall [28], também conhecido como Teorema de Hall. Foi primeiramente utilizado por Erdős, Rubin e Taylor [59] para provar que o grafo completo k -partido $K_{2,2,\dots,2}$ é k -selecionável, mostrando que $\chi_\ell(G) = \chi(G)$ se existe uma $\chi(G)$ -coloração em que cada classe de cor tem no máximo, dois vértices [80]. Em 1998, Maffray e Gravier [23] apresentaram algumas classes de grafos garras-livre em que o seu número cromático é igual ao seu número lista cromático, ou seja, $\chi(G) = \chi_\ell(G)$. A Tabela 4.2 apresenta os principais resultados da literatura de selecionabilidade em grafos que utilizam a técnica do sistema dos representantes distintos.

Seja S_1, S_2, \dots, S_k e k subconjuntos. Seja I um conjunto de m elementos, onde $I = \{a_1, a_2, \dots, a_k\}$, tal que os elementos a_i são todos distintos e tal que a_i pertence a S_i para cada $i = 1, \dots, k$. Então I é dito ser sistema dos representantes distintos para os subconjuntos S_1, S_2, \dots, S_k .

Teorema 4.3 (Hall [28]). *(Teorema dos Representantes distintos de Hall) Uma família de conjuntos S_1, S_2, \dots, S_m tem um sistema de representantes distintos se e só se para qualquer subconjunto $I \subseteq \{1, 2, \dots, m\}$ se verifica a condição*

$$|\bigcup_{i \in I} S_i| \geq |I|$$

Como exemplo, considere os conjuntos $S_1 = \{1, 2, 3\}$, $S_2 = \{1, 4, 5\}$ e $S_3 = \{3, 5\}$, então $I = \{1, 4, 5\}$ é um sistema de representantes distintos para S_1, S_2 e S_3 . O $I = \{1, 2, 5\}$ é outro SRD, porém $I = \{1, 3\}$ não é. Já os conjuntos $S_1 = \{2, 3, 4, 5\}$, $S_2 = \{4, 5\}$, $S_3 = \{4, 5\}$ e $S_4 = \{5\}$ não possuem sistema de representantes distintos. Agora considere $S_1 = \{1, 2\}$, $S_2 = \{1, 2\}$, $S_3 = \{1, 2\}$ e $S_4 = \{2, 3, 4, 5\}$ possui 5 (cinco) representantes distintos $I = \{1, 2, 3, 4, 5\}$, porém esse tipo de representação não é considerado, pois os três primeiros conjuntos não possuem representantes distintos para cada conjunto considerado.

Classe de Grafo	Características	Referências
k -partido K_{2*k}	k -seleccionável	[59]
k -partido $K_{3,3,2,\dots,2}$	Para $k \geq 3$, k -seleccionável	[66]
Livre de Guarras	$\chi_\ell(G) = \chi(G)$	[66]
k -partido K_{3*k}	$\lceil (4k - 1)/3 \rceil$ -seleccionável	[37]
k -partido $K_{4,2,2,\dots,2}$	k -seleccionável	[30]

Tabela 4.2: Resultados de selecionabilidade em grafos que utilizam a técnica de prova do sistemas dos representantes distintos.

4.2.3 Técnica de Thomassen

É uma técnica apresentada elegantemente por Carsten Thomassen [69]. Foi utilizada para provar que todo grafo planar é 5-seleccionável. O método propõe colorir alguns vértices que induzem um subgrafo conexo de um grafo G . Um limite é definido para incluir os vértices incolores e que são adjacentes aos vértices coloridos. Colore-se então os vértices do limite, ao qual é permitido que os mesmos tenham listas de cores de menor tamanho que os outros vértices de G . A prova que apresenta esta técnica é mostrada a seguir.

Para mostrar que todo grafo planar é o 5-seleccionável, a verificação será feita em grafos planares *quase*-triangulados. Um grafo **triangulado** é um grafo planar onde toda face é limitada por exatamente três arestas. Um grafo é **quase-triangular** se todas as faces são limitadas por exatamente três arestas, exceto a face externa (ou ilimitada). Um exemplo desse dois tipos de grafos triangulares são mostrados na Figura 4.12.

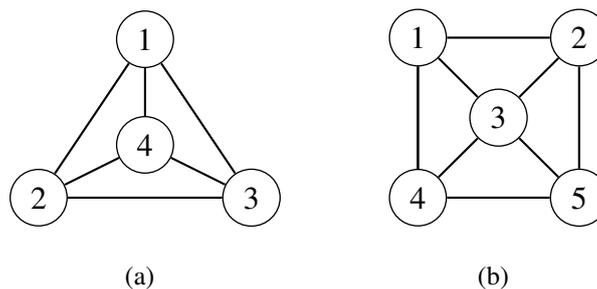


Figura 4.12: Exemplo de Grafos Triangulares. Em (a) temos um Grafo Triangulado e em (b) temos um Grafos *Quase*-Triangulado.

Teorema 4.4 (Thomassen [69]). *Todo grafo planar é 5-seleccionável.*

Prova. Assuma que G é um grafo *quase*-triangular e C é o ciclo que limita a face externa de G . Assuma também as seguintes colorações para G :

- a) Dois vértices adjacentes v_1 e v_2 de C são coloridos com as cores 1 e 2;
- b) $|L(v)| \geq 3$, para todo $v \in V(C)$;
- c) $|L(v)| \geq 5$, para todo $v \in V(G) - V(C)$.

Considere dois casos:

Caso 1. O ciclo C tem uma corda $\{uw\}$. O subgrafo G_1 é limitado por $C_1 \cup \{uw\}$ e contém os vértices v_1, v_2, u e w (observe a Figura 4.13) e, por hipótese de indução este subgrafo é 5-lista-coloração, pois é possível obter uma coloração própria com as listas de cores dadas.

Suponha que os vértices u e w recebam as cores 3 e 4. O subgrafo G_2 é limitado por $C_2 \cup \{uw\}$. Como v_1 e v_2 foram coloridos anteriormente, então podemos usar a hipótese de indução para G_2 . Por isso, o mesmo é verdade para todo o grafo G .

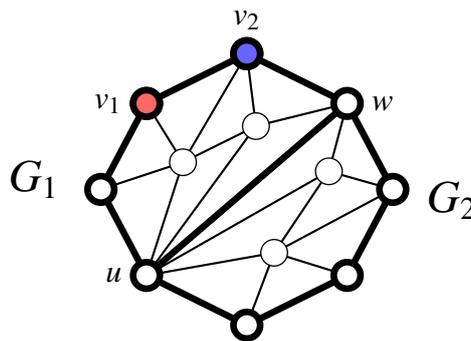


Figura 4.13: Exemplo de configuração de um grafo *quase*-triangular para a prova.

Caso 2. O ciclo C não possui corda. Seja G um grafo *quase*-triangulado. Seja v_k o vértice adjacente a v_1 e seja $v_1, u_1, u_2, \dots, u_t, v_{k-1}$ os vértices adjacentes a v_k . Com a retirada do vértice v_k , ou seja, $G' = G \setminus \{v_k\}$, as suas arestas adjacentes também são retiradas. Observe que na coloração proposta no caso (b), o ciclo externo C tem lista de cores $|L(v_i)| \geq 3$ e que existem dois vértices coloridos com as cores 1 e 2.

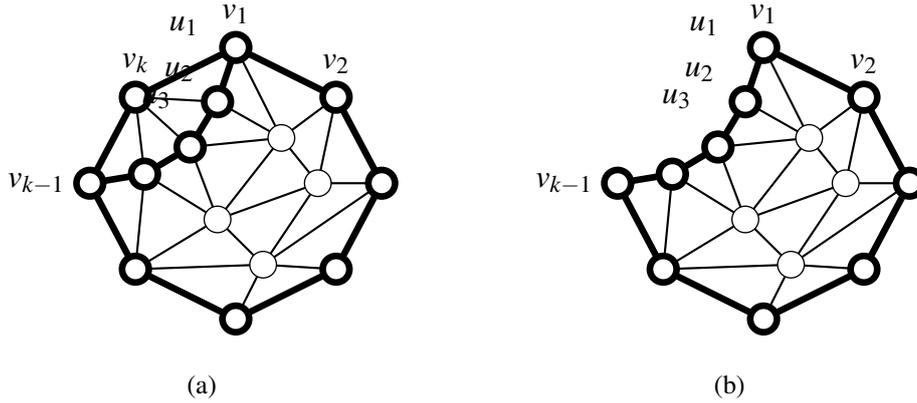


Figura 4.14: Exemplo de configuração de um grafo *quase*-triangular sem corda para a prova. (a) Grafo *quase*-triangular com o vértice v_k e (b) Grafo *quase*-triangular sem o vértice v_k e suas arestas adjacentes.

Como v_k pode receber as cores 1 ou 2, essas cores são eliminadas dos vizinhos de v_k , cada um ficando com $|L(u_i)| = 3$. Os vértices restantes de G' que não compartilham aresta com v_k permanecem com suas cores originais. O grafo G' satisfaz todas as três suposições iniciais e é, portanto, 5-lista-colorível por hipótese de indução. Ao retornar ao grafo original G , escolher a cor 1 ou 2, uma cor diferente de v_{k-1} e então G é 5-lista-colorível. \square

Alguns dos resultados na literatura que utilizam esse método foram apresentados na Tabela 4.5, que são: grafos planares K_5 -minor-livre e grafos planares com dois cruzamentos são 5-selecionáveis [64],[62] e grafos planares de circunferência no máximo 4 que são 3-selecionáveis [42].

4.2.4 Técnicas que envolvem digrafos

Existem duas técnicas de prova que envolvem digrafos, a técnica de kernel [80] (que não será tratada aqui) e a técnica de Alon-Tarsi [57]. Suponha uma orientação das arestas de um grafo G , de modo a formar um dígrafo D , e em seguida, atribua as listas de cores para os vértices, de tal maneira que cada vértice v tem uma lista $L(v)$ contendo mais cores do que o seu grau de saída em D .

$$|L(v)| \geq d^+(v) + 1 \quad (4.1)$$

Se D não possui ciclos direcionados então G é L -lista-colorível sempre que a Equação 4.1 é válida [80]. Contudo, há os casos em que existem ciclos direcionados nos digrafos. O método de prova proposto por Alon-Tarsi [57] trabalha com essas condições.

A técnica de Alon-Tarsi possui esse nome graças aos autores, Alon e Tarsi, e se utiliza da definição de grafos polinomiais [57][2]. Os mesmos utilizam esse método para mostrar que o grafo bipartido planar é 3-selecionável. Outros trabalhos que utilizaram esse método foram Fleischner e Stiebitz [20], para provar que se G é um grafo 4-regular sobre $3n$ vértices cujas arestas podem ser decompostas em um circuito hamiltoniano e n triângulos vértice disjuntos dois a dois, então $\chi_\ell(G) = \chi(G) = 3$ [80]. Já em [80], o autor utilizou a técnica para provar que em um grafo arco circular é válido que $\chi_\ell(G) = \chi(G)$.

4.3 Resultados em selecionabilidade para classes específicas de grafos

Esta seção apresenta os principais resultados da literatura de selecionabilidade em grafos em algumas classes de grafos. Existem diversos resultados em grafos planares, grafos bipartidos e multipartidos, além de resultados comparando classes de grafos em que o número cromático coincide com o número lista cromático.

4.3.1 Selecionabilidade em grafos bipartidos e multipartidos

Os primeiros resultados em selecionabilidade em grafos para grafos bipartidos foi apresentado por Erdős, Rubin e Taylor. Foi apresentado o clássico exemplo do grafo bipartido completo $K_{3,3}$ com listas de cores de tamanho dois para o qual o mesmo não é lista colorível. Também mostraram que dado um grafo bipartido completo $K_{m,m}$, se $m = \binom{2k-1}{k}$, então esse grafo não é k -selecionável [59]. A prova deste teorema segue abaixo:

Teorema 4.5 (Chartran [24]). *Se r e k são inteiros positivos tal que $r \geq \binom{2k-1}{k}$, então $\chi_\ell(K_{r,r}) \geq k + 1$.*

Prova. Assume-se, por contradição, que $\chi_\ell(K_{r,r}) \leq k$. Então existe uma k -lista-coloração de $K_{r,r}$. Seja U e W as partições de $K_{r,r}$, onde $U = \{u_1, u_2, \dots, u_r\}$ e $W = \{w_1, w_2, \dots, w_r\}$.

Seja $S = \{1, 2, \dots, 2k - 1\}$. Existem $\binom{2k-1}{k}$ subconjuntos distintos de k -elementos de S . Atribuir essas listas de cores para $\binom{2k-1}{k}$ vértices de U e para $\binom{2k-1}{k}$ vértices de W .

Os vértices remanescentes (se houver), aos mesmos poderão ser atribuídos qualquer dos subconjuntos de S . Para $i = 1, 2, \dots, r$, escolha a cor $a_i \in L(u_i)$ e seja $T = \{a_i : 1 \leq i \leq r\}$. Considere dois casos:

Caso 1. Seja $|T| \leq k - 1$. Então existe um subconjunto de k -elementos S' de S que é disjunto de T . Porém $L(u_j) = S'$ para algum j com $1 \leq j \leq r$. Uma contradição.

Caso 2. Seja $|T| \geq k$. Então existe um subconjunto de k -elementos T' de T . Assim, $L(w_j) = T'$ para algum j com $1 \leq j \leq r$. Não importa a cor que for atribuída para w_j , essa cor já foi atribuída para algum vértice u_i (observe o exemplo do $K_{3,3}$ na Figura 4.1). Assim, u_i e w_i possuem a mesma cor, sendo que $u_i w_i$ é uma aresta de $K_{r,r}$. Isso fere a restrição de que vértices adjacentes devem ter cores diferentes. Portanto, uma contradição. \square

Existem na literatura diversos resultados para grafos multipartidos completos. Seja um K_{m*k} um grafo k -partido completo com m vértices em cada partição. A Tabela 4.3 apresenta os diversos resultados para grafos multipartidos completos. Em 2002, Ohba *et al.* apresentaram alguns grafos multipartidos completo onde $\chi(G) \neq \chi_\ell(G)$. Kierstead [37] provou que todo grafos k -partido K_{3*k} é $\lceil (4k - 1)/3 \rceil$ -seleccionável. Em 1998, [23] mostrou que o grafo $K_{3\dots 3, 2*k}$ é k -seleccionável.

Tipos de Multipartidos	k -seleccionabilidade	Referências
$\chi_\ell(K_{2*k})$	k -seleccionável	[59]
$\chi_\ell(K_{3,2,\dots,2})$	k -seleccionável	[23]
$\chi_\ell(K_{3,3,2,\dots,2})$	k -seleccionável	[23]
$\chi_\ell(K_{4*k})$	$\lceil (3k - 1)/3 \rceil$ -seleccionável	[29]
$\chi_\ell(K_{3*k}) = k$	$\lceil (4k - 1)/3 \rceil$ -seleccionável	[37]
$\chi_\ell(K_{4,2,\dots,2})$	k -seleccionável, se k é ímpar, $k + 1$ -seleccionável, se k é par	[30]

Tabela 4.3: Resultados da k -seleccionabilidade para grafos multipartidos.

4.3.2 Seleccionabilidade em grafos planares

A classe de grafos planares têm sido um dos objetos de estudos principais em seleccionabilidade. De acordo com o Teorema das Quatro Cores, o número cromático de um grafo planar é no máximo 4 [36]. Erdős *et al.* [59] conjecturaram que a lista número

cromático de um grafo planar é no máximo 5 e que existem grafos planares que não são 4-selecionáveis. Esses resultados foram provados, respectivamente por Thomassen [69] e Voigt [74].

Resultados que verificam a não-selecionabilidade para um determinado k em grafos planares também são encontrados na literatura. Erdős *et al.* [59] conjecturaram que existiam grafos planares que não são 4-selecionável. Esse resultado foi provado por Voigt [74], com um grafo de ordem 238. Em 1996 Maryam Mirzakhani [53] deu um exemplo de um grafo planar de ordem 63, que não é 4-selecionável. A Tabela 4.4 apresenta os principais resultados da literatura de grafos planares que não são 4-selecionável.

Características	Referências
Grafos Planares que não são 4-selecionável	[74]
Grafos Planares sem C_3 que não são 3-selecionável	[75]
Grafos Planares que não são 4-selecionável	[76]
Grafos sem C_4 , C_5 e interseção de C_3 que não são 3-selecionáveis	[54]
Grafos Planares sem C_4 e C_5 que não são 3-selecionável	[77]

Tabela 4.4: Resultados da não k -selecionabilidade para grafos planares.

Existe na literatura diversos resultados que envolvem grafos planares, de acordo com algumas características presentes em alguns desses grafos, como grafos planares que não possuem um ciclo de tamanho sete (C_7) como subgrafo induzido [61] e grafos planares que não possuem diversos ciclos como subgrafo induzido, como o resultado de Thomassen [70], mostra que grafos planares sem C_3 e C_4 são 3-selecionável. A Tabela 4.5 apresenta os resultados encontrados na revisão da literatura.

k	Características	Referências
5	Todo grafo planar	[69]
	K_5 -menor-livre	[64]
	Planares com dois cruzamentos	[62]
4	Planar C_4 -livre	[61]
	Planar sem C_7	[61]
	Planar 2- C_3 -livre com um vértice em comum	[17]
3	Bipartido planar	[57]
	Planares de cintura 4 sem C_5 e C_6	[60]
	Planares de cintura 5	[70]
	Planar sem C_3 e C_4	[70]
	Planar sem i -ciclos para cada $i \in \{4, \dots, 9\}$	[6]
	Periplanares com cintura no máximo 4	[48]
	Planar sem C_4, C_5, C_6 ou C_9	[15]
	Planar sem C_4, C_6, C_7 ou C_9	[82]
	Planar sem i -ciclos para cada $i \in \{1, \dots, 10\}$ com uma corda	[16]
	Planar sem C_4, C_6, C_8 ou C_9	[44]
	Planar triângulo-livre sem C_3, C_6, C_7 e C_8	[46]
	Planar sem C_3, C_7 e C_8	[85]
	Alguns planares de cintura no máximo 4	[42]
	Planar triângulo-livre tal que C_4 não compartilha aresta com C_4 e C_5	[86]
	Planar sem C_4, C_5, C_8 e C_9	[83]
	Planar sem C_3 adjacentes e nem C_5, C_6 e C_9	[87]
Planar triângulo-livre tal que C_4 não é adjacente a C_4 ou C_5	[84]	

Tabela 4.5: Resultados da k -selecionabilidade para grafos planares.

4.3.3 Seleccionabilidade versus número cromático

Um grafo G é $\chi(G)$ -seleccionável se o seu número lista cromático coincide com seu número cromático. Como visto no início deste capítulo, isso não acontece em grafos bipartidos. Erdős *et al.* caracterizaram grafos em que $\chi_\ell(G) = \chi(G) = 2$, como o ciclo par e grafos $\theta_{2,2,2k}$. É fácil ver que ciclos ímpares o $\chi_\ell(G) = \chi(G) = 3$ [80]. Grafos árvores, que são 2-colorível, também são 2-seleccionável [24].

Aplicando a técnica de Hall (veja Seção 4.2.2), Erdős *et al.* [59] provaram que o grafo completo k -partido $K_{2,2,\dots,2}$ é k -seleccionável, mostrando que $\chi_\ell(G) = \chi(G)$ nesse caso. Gravier e Maffray [66] mostram que o complemento de um grafo triângulo-livre é válido que $\chi_\ell(G) = \chi(G)$. A Tabela 4.6 apresenta uma comparação de classes de grafos em que $\chi_\ell(G) = \chi(G)$ e $\chi_\ell(G) \neq \chi(G)$.

Classe de Grafos	k -coloração	Referências	k -seleccionável	Referências
Bipartido	2-colorível	[67]	3-seleccionável	[59]
Árvores	2-colorível	[67]	2-seleccionável	[24]
Ciclo Par	2-colorível	[80]	2-seleccionável	[24]
Ciclo Ímpar	3-colorível	[80]	3-seleccionável	[80]
Planar	4-colorível	[36]	5-seleccionável	[69]
Bipartido Planar	2-colorível	[67]	3-seleccionável	[57]
Planar sem C_3	3-colorível	[72]	4-seleccionável	[32]
Planar com cintura 5	3-colorível	[71]	3-seleccionável	[70]
Planar sem ciclos de C_4 a C_9	3-colorível	[6]	3-seleccionável	[43]
Periplanar Bipartido	2-colorível	[67]	3-seleccionável	[78]
$K_{2,2,\dots,2}$	k -colorível	[80]	k -seleccionável	[59]
Linha de um Bipartido	k -colorível	[58]	k -seleccionável	[58]
Garra-livre	k -colorível	[65]	k -seleccionável	[65]
Diamante	3-colorível	[12]	4-seleccionável	[47]
n -roda (n ímpar)	3-colorível	[12]	4-seleccionável	[47]
n -roda (n par)	4-colorível	[12]	4-seleccionável	[47]
{Garra, K_4 }-livre	6-colorível	[9]	4-seleccionável	[47]
Toroidal	4-colorível	[31]	4-seleccionável	[81]

Tabela 4.6: Tabela de comparação dos resultados existentes na literatura para k -coloração e k -seleccionabilidade em diversas classes de grafos. A parte sombreada indica que para essas classes de grafos, o resultado depende do valor de k .

Na literatura existe um teorema conhecido por Teorema de Nordhaus-Gaddum (Teorema 4.6), em que dado um grafo G de ordem n , então o seu número cromático mais o número cromático de seu complementar é menor ou igual a $n + 1$ [14].

Teorema 4.6. $\chi(G) + \chi(\bar{G}) \leq n + 1$

Na literatura existem alguns resultados em que o Teorema 4.6 atinge a igualdade. Alguns desses grafos foram caracterizados por Finck [18], que apresentou dois tipos de grafos com essa propriedade: Erdős *et al.* [59] apresentaram uma versão do Teorema 4.6 para lista coloração em grafos, como segue:

Teorema 4.7. $\chi_\ell(G) + \chi_\ell(\bar{G}) \leq n + 1$

Simone Dantas [13] provou a igualdade do Teorema 4.7 em algumas classes de grafos extremais, como por exemplo, um grafo extremal $F_2(S, K, C_5)$, onde seu conjunto de vértices pode ser particionado em uma clique K , um conjunto estável S e um C_5 tal que todo vértice do C_5 é adjacente a todo vértice de K e a nenhum vértice de S .

4.4 Algoritmo para determinar a k -selecionabilidade

Um algoritmo foi abordado para a k -selecionabilidade. O método enumera **todas** as listas de tamanho k para verificar se o grafo é colorível para todas essas listas e utiliza o Algoritmo 2 da Seção 3.3.2 para verificar se o grafo é colorível para cada conjunto de listas geradas.

O algoritmo inicia gerando dentro da função $\acute{e}_k\text{-selecionável}$, o conjunto de listas, formado pela combinação de elementos sem repetição, dado pela Equação 4.2, onde n equivale a quantidade de vértices do grafo tomados k a k . O valor inicial de k é 2, que é o valor que determina o tamanho das listas de cores.

$$X \leftarrow \binom{n}{k} \quad (4.2)$$

Formado os conjuntos de listas, o algoritmo forma os conjuntos de soluções de listas de cores, com elementos repetidos, ou seja, a permutação de listas de cores repetidas, dada pela função P_x , na linha 2 do Algoritmo 7.

Para cada conjunto $L \in P$ de listas para o grafo, o algoritmo faz uma chamada a função $Existe_Lista_Coloracao$ do Algoritmo 2 para verificar se o grafo possui coloração própria com este conjunto de listas. Se não for possível colorir, a função retorna *não* para a função $\acute{e}_k\text{-selecionável}$, onde o algoritmo incrementa k e recomeça a montagem de novas listas

de cores, agora com $k + 1$. Se for possível colorir, a função retorna *sim* para a função \acute{e}_k -selecionável, finalizando assim a execução do algoritmo.

Algoritmo 6: AlgoritmoGeral($G = (V, E)$)

```

1  $n \leftarrow |V|$ 
2  $k \leftarrow 2$ 
3 enquanto  $!(\acute{e}_k\text{-selecionável}(G, k))$  faça
4   |  $k \leftarrow k + 1$ 
5 fim enquanto
6 Retorne  $k$ 

```

Algoritmo 7: \acute{e}_k -selecionável($G = (V, E), k$)

```

1  $X \leftarrow \binom{n}{k}$ 
2  $P_x \leftarrow$  todas as  $n$ -permutações de  $X$  com elementos repetidos;
3 para cada  $L \in P_x$  faça
4   | se  $!(\text{Existe\_Lista\_Coloracao}(G, L, nao, 1))$  então
5     | Retorne NAO
6   | fim se
7 fim para cada
8 Retorne SIM

```

É importante enfatizar que o algoritmo efetua a verificação para **todos** os conjuntos de tamanho k . Quando o grafo não é k -selecionável, o algoritmo faz a verificação para o $k + 1$, e assim por diante, até encontrar um conjunto de listas para o qual o grafo seja colorível.

Observe o exemplo de execução dos Algoritmos 6 e 7 no grafo da Figura 4.15. As listas de cores possíveis são formadas pela Função 4.2, onde $n = 4$ e $k = 2$. Portanto, serão 6 conjuntos de listas de cores, tomados, 2 a 2, que são $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$, $\{2, 4\}$ e $\{3, 4\}$.

A seguir, uma possível solução é formado pelas listas anteriores, tomados n a n , onde pode-se existir listas repetidas. Um exemplo de permutação dos elementos do conjunto de listas apresentado seria:

$$L = (\{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\})$$

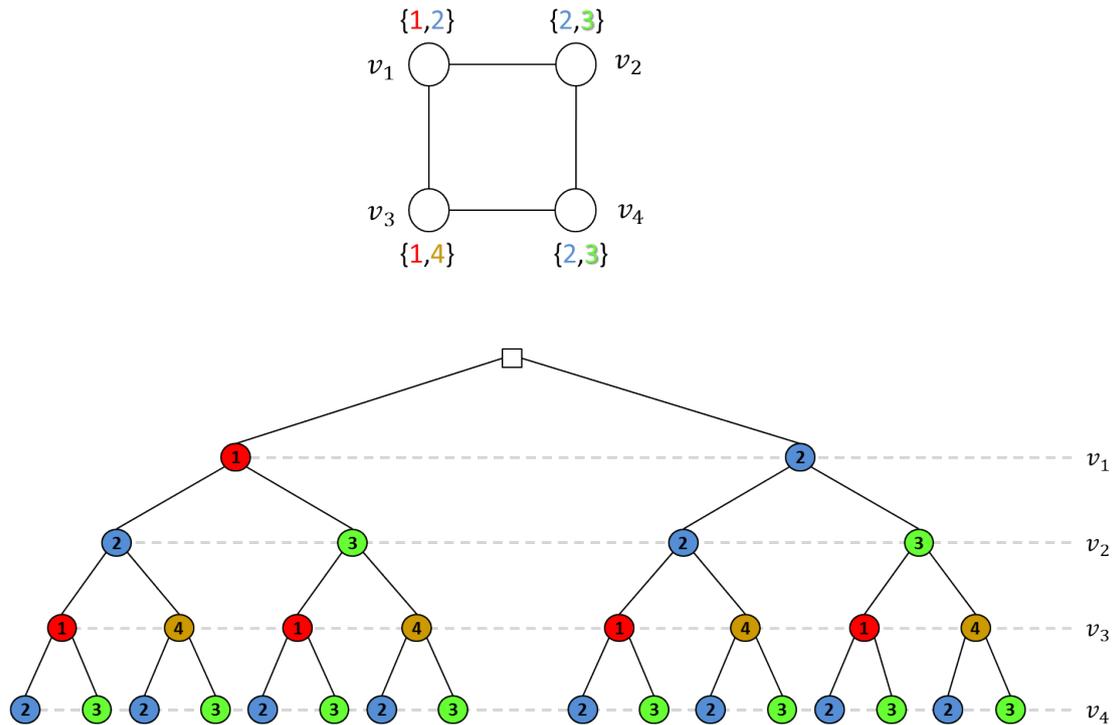


Figura 4.15: Exemplo de um grafo e árvore gerada pela execução do algoritmo para as listas de cores do grafo.

$$\begin{aligned}
 L &= (\{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 3\}) \\
 L &= (\{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 4\}) \\
 L &= (\{1, 2\}, \{1, 2\}, \{1, 2\}, \{2, 3\}) \\
 &\vdots \\
 L &= (\{1, 3\}, \{2, 3\}, \{1, 2\}, \{2, 4\})
 \end{aligned}$$

Para cada conjunto L de listas, é feita uma chamada ao Algoritmo 2 e assim, verificar se existe uma coloração própria para o conjunto de listas. No exemplo da Figura 4.15, a árvore de execução para as listas $L = (\{1, 3\}, \{2, 3\}, \{1, 2\}, \{2, 4\})$ é mostrado na Figura 4.15. Observe que no exemplo da árvore de execução, na Figura 4.15, o algoritmo da lista coloração, quando encontra uma solução, o restante da árvore de soluções para o grafo não precisa ser visitado, encerrando assim a execução do Algoritmo 2.

Um exemplo de execução dos Algoritmos 6 e 7 no grafo da Figura 4.15. As listas de cores possíveis são formadas pela Função 4.2, onde $n = 4$ e $k = 2$. Portanto, serão 6 conjuntos de listas de cores, tomados, 2 a 2, que são $p = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}$ e $\{3, 4\}\}$.

A seguir, as soluções são formadas pelas listas de cores anteriores, tomados n a n , onde

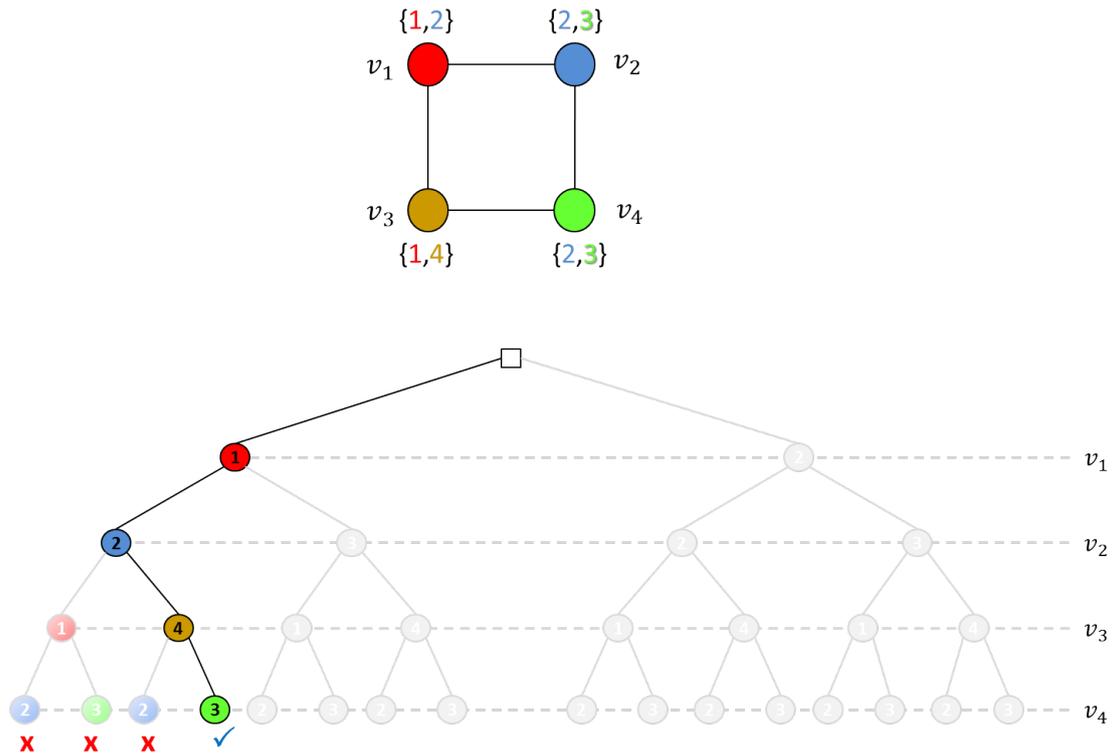


Figura 4.16: Exemplo de solução encontrada com a execução do algoritmo para selecionabilidade em grafos.

é possível elementos repetidos. Essa notação é dada por AR_p^n . O princípio multiplicativo diz que o número total de maneiras de se retirar n elemento dos p objetos, distintos ou não, é igual a

$$AR_p^n = p^n \tag{4.3}$$

uma vez que o primeiro elemento pode ser retirado de p maneiras, o segundo também de p maneiras, e assim sucessivamente, até que o n -ésimo seja escolhido.

Voltando ao caso da geração de soluções de listas de cores, o grafo da Figura 4.15 possui 4 vértices e 6 listas de cores de tamanho 2 (relembrando $\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}$ e $\{3,4\}$). No caso, as listas de cores serão tomados n a n , podendo ter listas repetidas. Logo, o cálculo será feito de acordo com a fórmula da Equação 4.3 para calcular o número de todas as soluções de listas de cores. Considerando as variáveis da equação, m representa o número de vértices do grafo e n representa a quantidade de listas de cores:

$$p^n = 6^4 = 1296 \tag{4.4}$$

Ou seja, para o grafo da Figura 4.15, com 6 possíveis listas de cores de tamanho 2, temos 1296 soluções de lista coloração.

4.5 Considerações do capítulo

Este capítulo apresentou as principais definições em selecionabilidade em grafos, além de apresentar com detalhes as provas dos principais resultados existentes na literatura, como a propriedade em grafos bipartidos completos e grafos planares. Além disso, um algoritmo em k -selecionabilidade foi apresentado com o seu respectivo cálculo da geração das listas de cores.

Capítulo 5

Contribuições

Neste capítulo é introduzido a k - (γ, μ) -selecionabilidade, uma correlação da k -selecionabilidade e da (γ, μ) -coloração. São apresentadas as provas em k - (γ, μ) -selecionabilidade em duas classes de grafos, os grafos periplanares, que são 3- (γ, μ) -selecionável e grafos bipartidos completos, que são 2- (γ, μ) -selecionável, além de uma prova mais geral, de grafos que são k -coloríveis são k - (γ, μ) -coloríveis. Ao final, o algoritmo abordado no capítulo anterior para k -selecionabilidade será abordado também para k - (γ, μ) -selecionabilidade, comparando a quantidade de listas de cores geradas por ambos os algoritmos.

5.1 Definição da propriedade da k - (γ, μ) -selecionabilidade

Nos resultados principais, a (γ, μ) -coloração foi correlacionada com a k -selecionabilidade, onde dado um grafo G , a mesma é atribuído a seus vértices listas de cores do tipo (γ, μ) e todas essas listas de cores possuem tamanho k . A definição desta coloração será dada a seguir:

Definição 5.1. *Um grafo G é k - (γ, μ) -selecionável ou k - (γ, μ) -lista-colorível se G é (γ, μ) -colorível para cada coleção \mathcal{L} de listas $L(v)$ para os vértices v de G tal que $|L(v)| \geq k$ para cada vértice v , e as listas são do tipo (γ, μ) .*

O menor inteiro k para o qual G é k - (γ, μ) -lista-colorível para todas as listas de cores tipo (γ, μ) de tamanho k é denominado número lista cromático γ, μ , também representado

por $\chi_{\ell,(\gamma,\mu)}$. A Figura 5.1 apresenta o K_3 , um grafo que é 3-colorível, 3-selecionável e também 3- (γ,μ) -selecionável. Ou seja, o K_3 é colorível para qualquer lista de cores de tamanho três do tipo γ,μ atribuído à seus vértices.

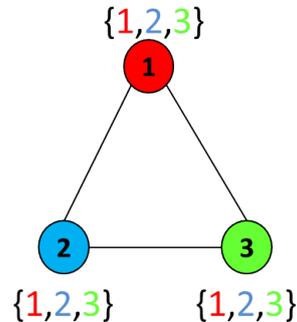


Figura 5.1: Grafo K_3 em que o $\chi_{\ell,(\gamma,\mu)} = 3$.

5.2 Aplicação das técnicas de provas de selecionabilidade em k - (γ,μ) -selecionabilidade

Esta seção apresenta os principais resultados obtidos para a k - (γ,μ) -selecionabilidade, utilizando algumas das técnicas de provas apresentadas no Capítulo 4. Será apresentado uma prova de que grafos periplanares, que são 2-degenerados [45], e portanto são 3- (γ,μ) -selecionável.

5.2.1 k - (γ,μ) -selecionabilidade em grafos periplanares

Esta seção apresenta a prova de que grafos periplanares são 3- (γ,μ) -selecionável. Em 1972, Lick e White [45] apresentaram a caracterização de k -degeneração em diversas classes de grafos, dentre elas, o **grafo periplanar**, que é 2-degenerado. Para mostrar que o grafo periplanar é 2-degenerado, é suficiente mostrar que todo subgrafo de grafo periplanar é periplanar e todo grafo periplanar não trivial contém pelo menos 2 vértices de grau 2 ou menos [45].

Teorema 5.1 (Chartrand [24]). *Todo grafo periplanar não trivial contém pelo menos 2 vértices de grau 2 ou menos.*

Prova. Seja G um grafo periplanar não trivial. Esse resultado é trivial se G é de ordem 4 ou menos. Logo, assume-se que G é de ordem 5 ou mais. Adicionando arestas em G , se necessário, para obter um grafo periplanar maximal. Assim, o limite da região exterior de G é um ciclo hamiltoniano C .

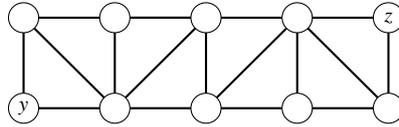


Figura 5.2: Grafo periplanar maximal para demonstração do Teorema 5.2.1.

Dentre as cordas de C , seja uv tal que uv e um caminho $u - v$ em C produz um ciclo que contém um número mínimo de regiões. Necessariamente, esse mínimo é 1. Então, o grau do vértice remanescente y do limite dessa região é 2. Existe uma corda wx de C por outro caminho $u - v$ de C que produz um vértice z de grau 2. Em G (na Figura 5.2), os graus de y e z são, portanto, 2 ou menos. \square

Teorema 5.2. *Todo grafo periplanar é $3-(\gamma, \mu)$ -selecionável.*

Prova. A prova é feita por indução sobre n . Seja $G = K_1$. Este grafo não tem arestas e é trivialmente periplanar e $3-(\gamma, \mu)$ -selecionável.

Supondo que todo grafo periplanar com n vértices é $3-(\gamma, \mu)$ -selecionável, será demonstrado que a lista coloração também é válido para um grafo periplanar G_{n+1} com $n + 1$ vértices e com uma distribuição de listas de cores em seus vértices do tipo (γ, μ) de tamanho 3. Pelo Teorema 5.1, existe um vértice v em G_{n+1} com grau até no máximo 2. Removendo este vértice, tem-se um subgrafo G_n com n vértices.

Uma vez que G_{n+1} pode ser desenhado com seus vértices em forma de ciclo com cordas sem cruzamentos (planar), o subgrafo também pode ser desenhado dessa forma. Isso implica que G_n tem uma lista coloração com listas do tipo (γ, μ) de tamanho 3. Colorir os vértices de G_{n+1} com 3 ou mais cores. Uma vez que os vértices adjacentes ao vértice v de grau no máximo 2 já estão coloridos, considere dois casos:

- O vértice v possui grau um. Nesse caso, o vértice v recebe uma cor da lista de cores e o vértice adjacente recebe outra cor. Como a lista de cores tem tamanho três, uma cor sobrar.

- O vértice v possui grau dois. O vértice v receberá uma terceira cor da sua lista de cores que seja diferente de seus adjacentes. E assim a demonstração é válida.

□

5.2.2 k - (γ, μ) -selecionabilidade em grafos bipartido completo

Será apresentado a prova para o grafo bipartido completo, que é 3-selecionável [59], porém é 2- (γ, μ) -selecionável. Antes, será apresentado as provas de alguns lemas importantes para a prova principal.

Lema 5.1. *Considere um grafo bipartido completo $K_{n,n} = G[V, W]$ para $n \geq 1$ com partições $V = \{v_1, \dots, v_n\}, W = \{w_1, \dots, w_n\}$ e uma distribuição de listas (γ, μ) de tamanho 2, $L(v)$ para todo $v \in K_{n,n}$ tal que $L(v_i) \neq L(v_j)$ se $i \neq j$, e $L(v_i) = L(w_i)$ para todo $i = 1, \dots, n$. Então o $K_{n,n}$ é 2- (γ, μ) -selecionável.*

Prova. Será provado por indução sobre n . Para $n = 1$, o grafo $K_{1,1}$ possui dois vértices v_1 e w_1 e $L(v_1) = L(w_1) = \{1, 2\}$. Desta forma escolhe-se a cor 1 para o vértice v_1 e a cor 2 para o vértice w_1 , e têm-se uma lista coloração para o $K_{1,1}$.

Suponha agora que o teorema é válido para $K_{n,n}$, onde $n \geq 1$ e então vale para o $K_{n+1, n+1}$. Sem perda de generalidade, supor que as listas $L(v_i) = \{\gamma_i, \gamma_i + 1\}$ estão distribuídas nos vértices tal que $\gamma_1 < \gamma_2 < \dots < \gamma_{n+1}$. Considere o subgrafo completo $K_{n,n}$ obtido pelos vértices v_1, v_2, \dots, v_n e w_1, w_2, \dots, w_n . Esse grafo está representado pela Figura 5.3.

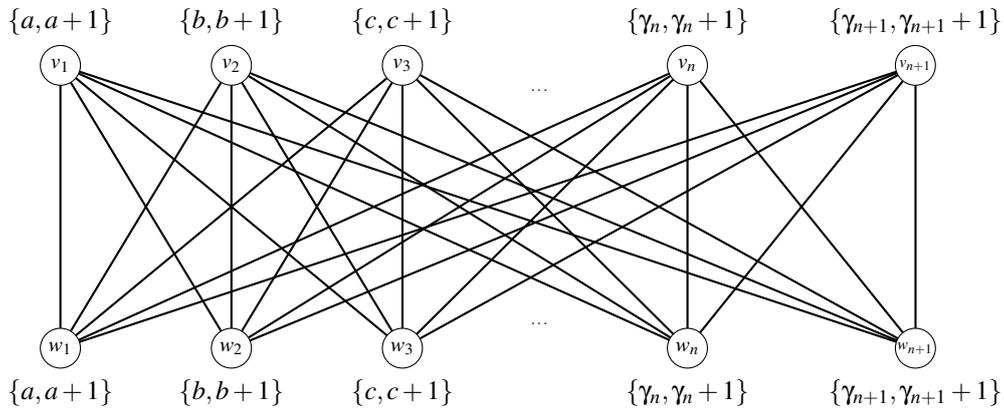


Figura 5.3: Grafo $K_{n,n}$ com partições com listas de cores iguais.

Por hipótese de indução existe uma lista coloração para este $K_{n,n}$. Falta colorir os vértices v_{n+1} e w_{n+1} . No pior dos casos $\gamma_{n+1} = \gamma_n + 1$ pois $\gamma_n < \gamma_{n+1}$ e as listas são

em sequência. Na Figura 5.4 há um exemplo de $K_{3,3}$, onde o pior caso, dois vértices contém a lista de cores $\{5, 6\}$ e esse dois vértices estão representados pelos vértices v_{n+1} e w_{n+1} . Neste caso então, se for $\gamma_{n+1} = \gamma_n + 1$, sem perda de generalidade, supor que a cor escolhida para o vértice v_n foi γ_n e para o vértice w_n foi $\gamma_n + 1$. Então o vértice v_{n+1} será colorido com a cor $\gamma_{n+1} + 1$ e o vértice w_{n+1} com a cor γ_{n+1} .

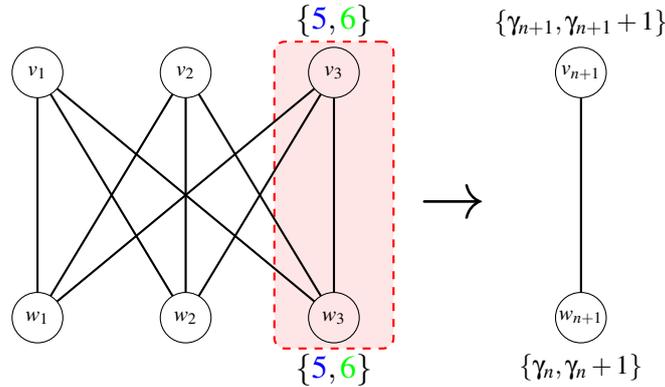


Figura 5.4: Exemplo de um grafo $K_{3,3}$. O subgrafo com uma linha tracejada representa o subgrafo com os vértices v_{n+1} e w_{n+1} .

Se $\gamma_{n+1} < \gamma_n + 1$ não é possível fazer esta mesma coloração para os vértices v_{n+1} e w_{n+1} . Essa coloração pode ser observada na Figura 5.5, onde o vértice v_n é colorido com a cor γ_n (cor vermelha, 4), o vértice v_{n+1} é colorido com a cor $\gamma_{n+1} + 1$ (cor verde, 6), o vértice w_n e o vértice w_{n+1} são coloridos com a cor γ_{n+1} e $\gamma_{n+1} + 1$ (cor azul, 5). Portanto $K_{n+1,n+1}$ é $2-(\gamma, \mu)$ -selecionável.

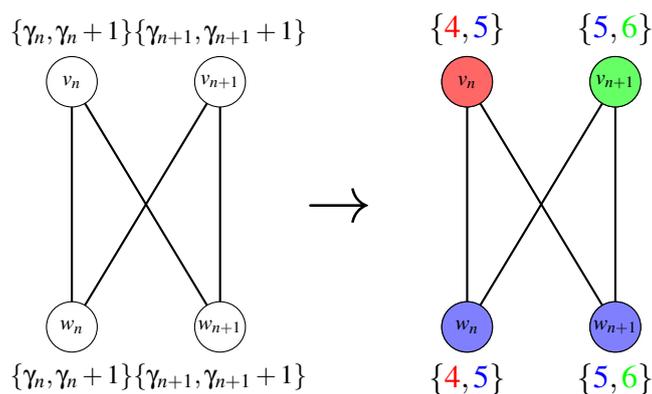


Figura 5.5: Subgrafo de um $K_{n,n}$. Coloração feita nos vértices v_{n+1} e w_{n+1} .

□

Lema 5.2. Considere um $K_{n,n} = G[V, W]$ para $n \geq 1$ com partições $V = \{v_1, \dots, v_n\}$, $W =$

$\{w_1, \dots, w_n\}$ e uma distribuição de listas (γ, μ) de tamanho 2, $L(v)$ para todo $v \in K_{n,n}$ tal que $L(v_i) \neq L(w_j)$ para todo $i, j = 1, \dots, n$. Então o $K_{n,n}$ é $2 - (\gamma, \mu)$ -seleccionável.

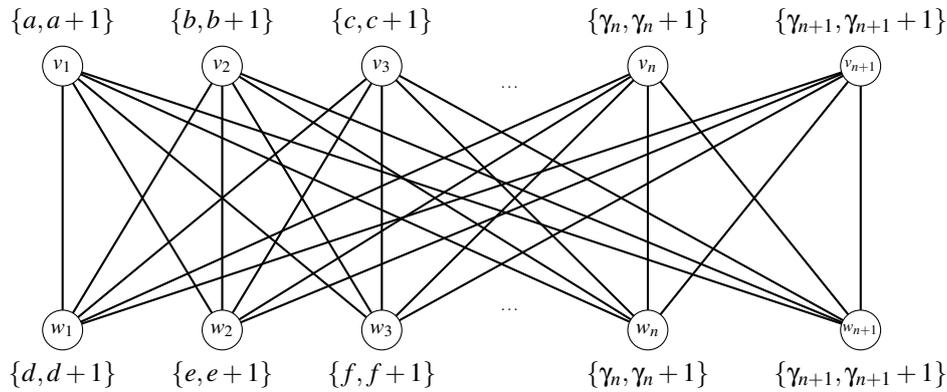


Figura 5.6: Grafo $K_{n,n}$ com partições com listas de cores diferentes.

Prova. Considere na partição V somente as listas distintas S_1, \dots, S_r , e na partição W somente as listas distintas S_{r+1}, \dots, S_{r+t} . Assim, $S_i \neq S_j$, para $i \neq j$, e como são listas (γ, μ) de tamanho 2, a interseção de quaisquer duas destas listas tem no máximo um elemento. Assim, estes conjuntos S_i cumprem o Teorema 4.3 e assim têm-se elementos distintos $c_i \in S_i$, com $c_i \neq c_j$. Então colore-se um vértice $v \in K_{n,n}$ com a cor c_i se $L(v) = S_i$, assim como é feita na Figura 5.7, e assim uma lista coloração do $K_{n,n}$ é realizada.

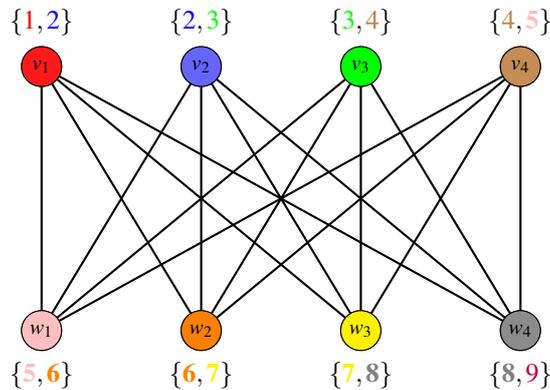


Figura 5.7: Grafo $K_{4,4}$. Cada vértice foi colorido com uma cor distinta c_i tal que $L(v) = S_i$, cumprindo o Teorema 4.3.

□

Teorema 5.3. O grafo bipartido completo $K_{n,n}$ é $2 - (\gamma, \mu)$ -seleccionável para $n \geq 1$.

Prova. Seja $G = K_{n,n}$ um grafo bipartido completo com partições $V = \{v_1, v_2, \dots, v_n\}$ e $W = \{w_1, w_2, \dots, w_n\}$, para $n \geq 1$. Considere uma distribuição de listas de cores do tipo

(γ, μ) de tamanho 2, $L(v)$ para todo vértice v de G . Escolha as listas de cores que aparecem ao mesmo tempo nos vértices das duas partições V e W . Claramente algumas dessas listas podem aparecer em mais de um vértice em cada partição, porém considere apenas uma lista representante no caso dela aparecer mais de uma vez numa mesma partição. Deste modo, considere os conjuntos S_1, S_2, \dots, S_r ($r \leq n$) (Figura 5.8), que são as listas que ocorrem ao mesmo tempo em vértices de V e W , e como foi selecionado apenas um representante em caso de repetição na partição, tem-se que $S_i \neq S_j$, para $i \neq j$.

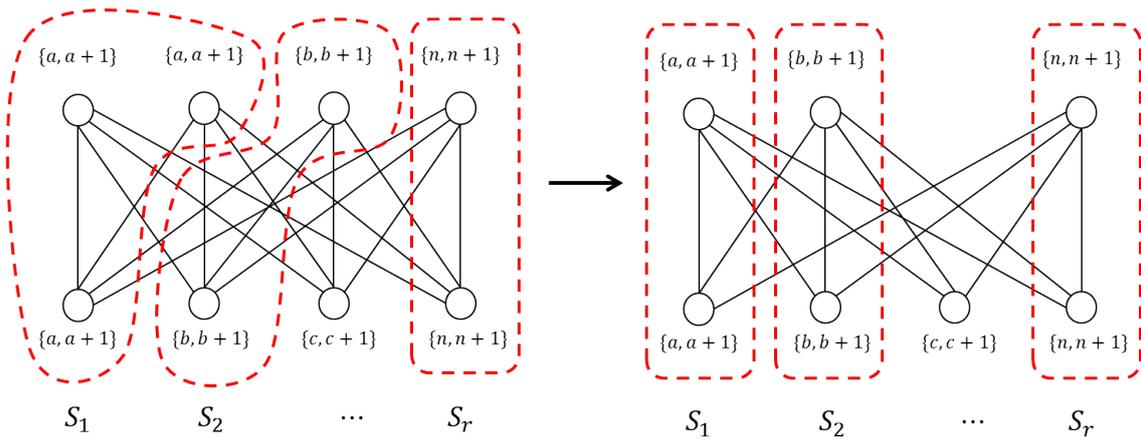


Figura 5.8: Para $r \geq 1$, temos os conjuntos S_1, S_2, \dots, S_r .

Se $r = 0$, o grafo G cumpre as hipóteses do Lema 5.2 e assim existe uma lista coloração de G .

Se $r \geq 1$, formar um subgrafo completo $K_{r,r}$ com estas listas de cores S_1, S_2, \dots, S_r e um vértice representante para cada uma delas. Um exemplo desta configuração é dado na Figura 5.9. Este grafo $K_{r,r}$ cumpre as hipóteses do Lema 5.1 e portanto existe uma lista coloração para este subgrafo $K_{r,r}$. Sejam então x_1, x_2, \dots, x_r as cores escolhidas para estes r vértices de V e y_1, y_2, \dots, y_r as cores escolhidas para estes r vértices de W . Estende-se estas cores para todos os vértices restantes de G que possuem uma destas listas de cores S_1, S_2, \dots, S_r , colorindo com a cor x_i para os demais vértices de V que possuem a lista S_i e com a cor y_i para os demais vértices de W que possuem a lista S_i .

Agora, os vértices v de G que ainda não foram coloridos possuem listas de cores $L(v)$ que estão apenas numa partição de G . Ainda assim, em cada partição V e W podem haver listas que se repetem, ou seja, mais de um vértice podem ter a mesma lista em uma mesma partição. Deste modo, novamente considere apenas uma lista representante de cada uma dessas restantes, definindo os conjuntos: E_1, E_2, \dots, E_t , que são as listas distin-

tas distribuídas nos vértices restantes de V , e F_1, F_2, \dots, F_s , as listas distintas distribuídas nos vértices restantes de W , observe a Figura 5.10.

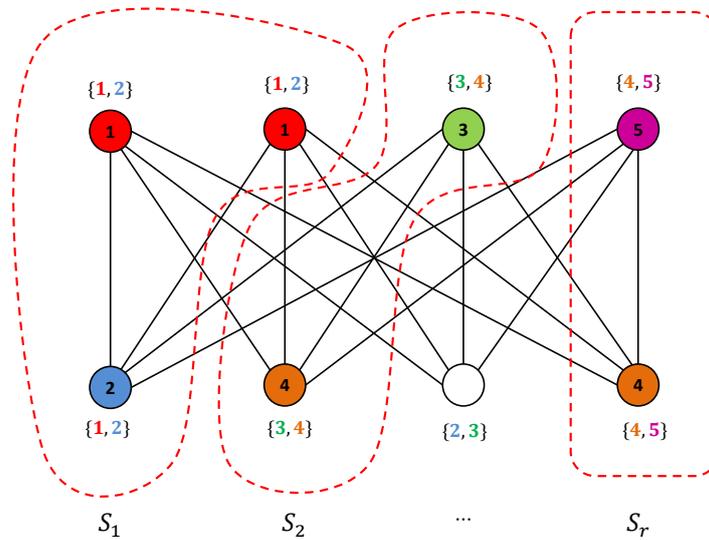


Figura 5.9: Conjuntos S_1, S_2, \dots, S_r coloridos.

Por construção, considere que $E_i \neq E_j$ para $i \neq j$, $F_i \neq F_j$ para $i \neq j$ e $E_i \neq F_j$, $\forall i, j$. Além disso, por serem listas de tamanho 2 do tipo (γ, μ) duas dessas listas distintas possuem no máximo um elemento em comum.

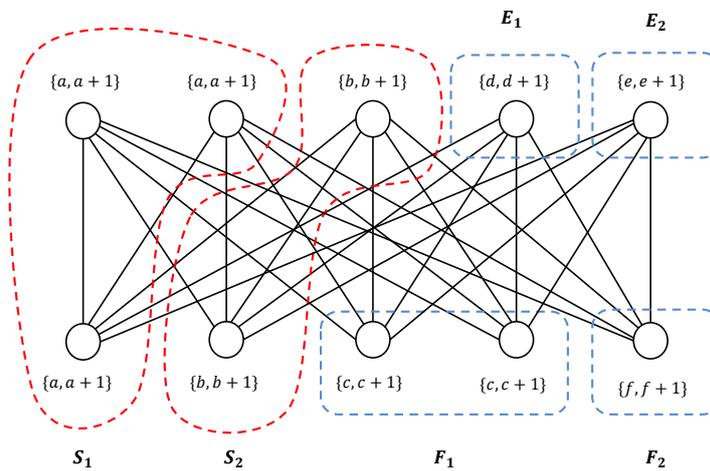


Figura 5.10: Conjuntos E_1, E_2, \dots, E_t e F_1, F_2, \dots, F_s .

Considere então somente as cores distintas dentre as cores $x_1, x_2, \dots, x_r, y_1, y_2, \dots, y_r$. Pela coloração é claro que $x_i \neq y_j, \forall i, j$. Defina estas cores distintas como sendo $x_{r_1}, x_{r_2}, \dots, x_{r_p}, y_{r_1}, y_{r_2}, \dots, y_{r_u}$.

Defina então os conjuntos $A_1 = \{x_{r_1}\}, A_2 = \{x_{r_2}\}, \dots, A_p = \{x_{r_p}\}, A_{p+1} = \{y_{r_1}\}, A_{p+2} = \{y_{r_2}\}, \dots, A_{p+u} = \{y_{r_u}\}, A_{p+u+1} = E_1, \dots, A_{p+u+t} = E_t, A_{p+u+t+1} = F_1, \dots, A_{p+u+t+s} = F_s$. Como as listas E_i e F_j são do tipo (γ, μ) de tamanho 2 e distintas, temos que $|A_i \cap A_j| \leq 1$, ou seja, $3 \leq |A_i \cup A_j| \leq 4, \forall i \neq j \in \{p+u+1, \dots, p+u+t+s\}$.

Cada conjunto A_i , para $i \geq p+u+1$, pode contar no máximo um dos elementos $x_{r_1}, x_{r_2}, \dots, x_{r_p}, y_{r_1}, y_{r_2}, \dots, y_{r_u}$. Assim,

$$3 \leq |A_i \cup \{x_{r_j}\}| \leq 4, \quad i \geq p+u+1, j = 1, \dots, p$$

$$3 \leq |A_i \cup \{y_{r_j}\}| \leq 4, \quad i \geq p+u+1, j = 1, \dots, u$$

$$3 \leq |A_i \cup \{x_{r_j}\} \cup \{y_{r_k}\}| \leq 4, \quad i \geq p+u+1, j = 1, \dots, p, k = 1, \dots, u.$$

Assim, dado qualquer $I \subseteq \{1, \dots, p+u+t+s\}$ tem-se que $|\bigcup_{i \in I} A_i| \geq |I|$, e pelo Teorema 4.3 que garante a escolha de elementos distintos $c_i \in A_i$, para $i = 1, \dots, p+u+t+s$, com $c_i \neq c_j$, para $i \neq j$. Assim, escolha as cores $c_i, i = p+u+1, \dots, p+u+t+s$ para os vértices que não estavam coloridos de tal forma que o vértice receberá a cor c_i se sua lista de cores for $A_i, i = p+u+1, \dots, p+u+t+s$, completando assim a lista coloração de $K_{n,n}$ e concluindo a prova. □

5.3 Grafos k -coloríveis são k - (γ, μ) -selecionáveis

A seguir, será apresentado uma prova mais geral envolvendo k - (γ, μ) -selecionabilidade. Dado um grafo G , se G é k -colorível então G é k - (γ, μ) -selecionável.

Teorema 5.4. *Seja G um grafo k -colorível. Então G é k - (γ, μ) -selecionável.*

Prova. Como G é k -colorível, existe uma coloração $c : V \rightarrow \mathbb{N}$ tal que $c(v) \in \{0, 1, 2, \dots, k-1\}$, e se os vértices v_i e v_j são adjacentes $c(v_i) \neq c(v_j)$.

Em uma divisão de um inteiro qualquer m por k , obtêm-se o resto da divisão r , tal que $r \in \{0, 1, 2, \dots, k-1\}$. Além do mais esta divisão particiona os inteiros em k conjuntos distintos. Considere $A_0 = \{n \in \mathbb{Z} \mid \text{o resto da divisão de } n \text{ por } k \text{ é } 0\}$, $A_1 = \{n \in \mathbb{Z} \mid \text{o resto da divisão de } n \text{ por } k \text{ é } 1\}$, $A_2 = \{n \in \mathbb{Z} \mid \text{o resto da divisão de } n \text{ por } k \text{ é } 2\}$, \dots ,

$A_{k-1} = \{n \in \mathbb{Z} \mid \text{o resto da divisão de } n \text{ por } k \text{ é } k-1\}$. Estes conjuntos são dois a dois disjuntos, ou seja, $A_i \cap A_j = \emptyset$, se $i \neq j$.

Outro fato importante é que dada qualquer sequência de k inteiros consecutivos $x, x+1, x+2, \dots, x+k-1$, existe exatamente um destes elementos contido em cada um dos conjuntos A_i . Ou seja, um desses elementos deixa resto i na divisão por k , para $i = 0, 1, 2, \dots, k-1$.

Considere agora uma distribuição qualquer de listas de tamanho k do tipo (γ, μ) para os vértices de V . Assim, para cada $v_i \in V$, $L(v_i) = \{\gamma_i, \gamma_i + 1, \dots, \gamma_i + k - 1\}$. Seja v_i e v_j vértices adjacentes em G , assim $c_i = c(v_i) \neq c(v_j) = c_j$, e portanto os conjuntos A_{c_i} e A_{c_j} são disjuntos. Para uma (γ, μ) -lista coloração de G nestes vértices adjacentes v_i e v_j escolha como cor para o vértice v_i o elemento de $L(v_i)$ que pertence ao conjunto A_{c_i} e para o vértice v_j escolhamos o elemento de $L(v_j)$ que pertence ao conjunto A_{c_j} e isto garante que vértices adjacentes serão coloridos com cores distintas. Portanto G é k - (γ, μ) -selecionável. \square

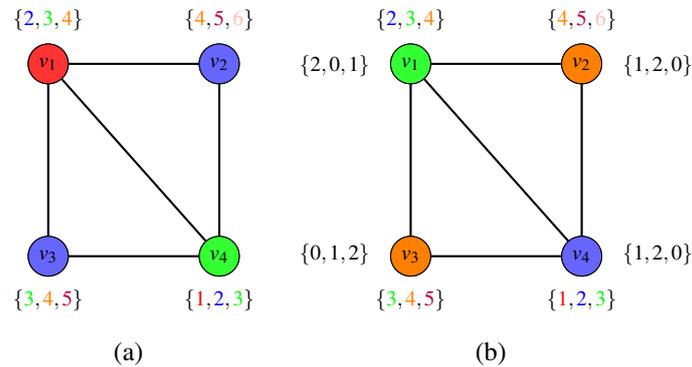


Figura 5.11: Uma k - (γ, μ) -coloração de um grafo, que é 3-colorível, portanto é 3- (γ, μ) -selecionável.

A Figura 5.11 apresenta um exemplo da coloração da prova. A Figura 5.11(a) apresenta um grafo 3-colorido (considere 0 vermelho, 1 azul e 2 verde) com uma distribuição qualquer de lista de cores do tipo (γ, μ) . A Figura 5.11(b) apresenta ao lado de cada vértice uma lista do resultado do resto da divisão das listas de cores de cada vértice por 3. A cor à ser escolhida para o vértice será a qual o resto da divisão dessa cor por 3 seja igual a cor já atribuída ao vértice. No exemplo, as cores escolhidas são: $v_1 = 3$, $v_2 = 4$, $v_3 = 4$ e $v_4 = 2$.

5.4 Algoritmos para determinar a $k-(\gamma, \mu)$ -selecionabilidade

Nesta seção, serão abordados os algoritmos implementados para $k-(\gamma, \mu)$ -selecionabilidade. O primeiro é um algoritmo exato para encontrar a k -coloração de um grafo. Esse algoritmo se fez necessário, com base na prova do Teorema 5.3, que afirma que todo grafo que é k -colorível é $k-(\gamma, \mu)$ -colorível. Desse algoritmo, podemos colorir o grafo com listas do tipo γ, μ de tamanho k facilmente. O segundo algoritmo abordado retrata um método exato para $k-(\gamma, \mu)$ -selecionabilidade, utilizando o algoritmo da k -coloração para determinar o tamanho das listas de cores que serão geradas. Esses algoritmos serão apresentados em detalhes nas próximas seções.

O primeiro algoritmo abordado utiliza o método de enumeração implícita para encontrar a k -coloração de um grafo, e que foi implementado para $k-(\gamma, \mu)$ -selecionabilidade. Com base no Teorema 5.3, todo grafo que tem uma k -coloração é $k-(\gamma, \mu)$ -colorível. Portanto, basta um algoritmo para encontrar a k -coloração que encontraremos a $k-(\gamma, \mu)$ -coloração.

O Algoritmo 19 para encontrar a k -coloração de um grafo G , inicia na Linha 1, verificando se o nível da árvore de recursão é igual ao número de vértices n do grafo, ou seja, se todos os vértices do grafo estão coloridos. Como no primeiro instante essa verificação é falsa, o algoritmo segue para a Linha 6, para iniciar a coloração.

O vértice inicialmente é marcado com *SIM*, para informar que o mesmo foi colorido. Logo após, o vértice receberá uma cor na estrutura $col \leftarrow col \cup (v, k)$. Antes de avançar para outro vértice, o algoritmo verifica a factibilidade da cor atribuída ao vértice atual na Linha 11. Se não for factível, então o algoritmo testa outra cor. Se for factível, o algoritmo segue para a recursão para o próximo vértice. Quando todos os vértices estão coloridos, temos uma coloração própria do grafo, porém, falta verificar se a coloração do grafo é a menor. Esse passo é realizado na Linha 2.

Um exemplo da execução do Algoritmo 19 é apresentado na Figura 5.12, que possui um grafo K_3 e a árvore de execução do algoritmo. Na árvore de execução, no final de cada nível, o **X** indica que a coloração para o grafo do exemplo é inválida. Já o \checkmark indica que a

coloração é válida.

Algoritmo 8: k -coloração($G = (V, E), \text{col}, \text{jacol}, \text{nivel}$)

```
1 se todos os vértices foram coloridos então
2   se a coloração própria é mínima então
3     É a melhor solução
4   fim se
5 senão
6   para cada  $v \in V$  faça
7     se ( $\text{cor}(v) == \text{NÃO}$ ) então
8        $\text{cor}(v) \leftarrow \text{SIM}$ 
9       para cada  $k \leftarrow 0$  até  $n - 1$  faça
10         $\text{col} \leftarrow \text{col} \cup (v, k)$ 
11        se a coloração não é factível então
12          Testa a próxima cor
13        fim se
14         $k$ -coloração( $G, \text{col}, \text{jacol}, \text{nivel}+1$ );
15         $\text{col} \leftarrow \text{col} - (v, k)$ ;
16      fim para cada
17    fim se
18  fim para cada
19 fim se
```

O algoritmo começa atribuindo as cores para os vértices do grafo. Quando uma das cores atribuídas fere a restrição de coloração, o algoritmo não desce mais na árvore e continua para uma outra solução. Essa *não* descida na árvore é indicada pelo **X**. Esse método de *não descida* na árvore *poda* as soluções infrutíferas. Quando o algoritmo encontra uma solução válida, a melhor solução é guardada (esse passo está na Linha 2 do Algoritmo 19).

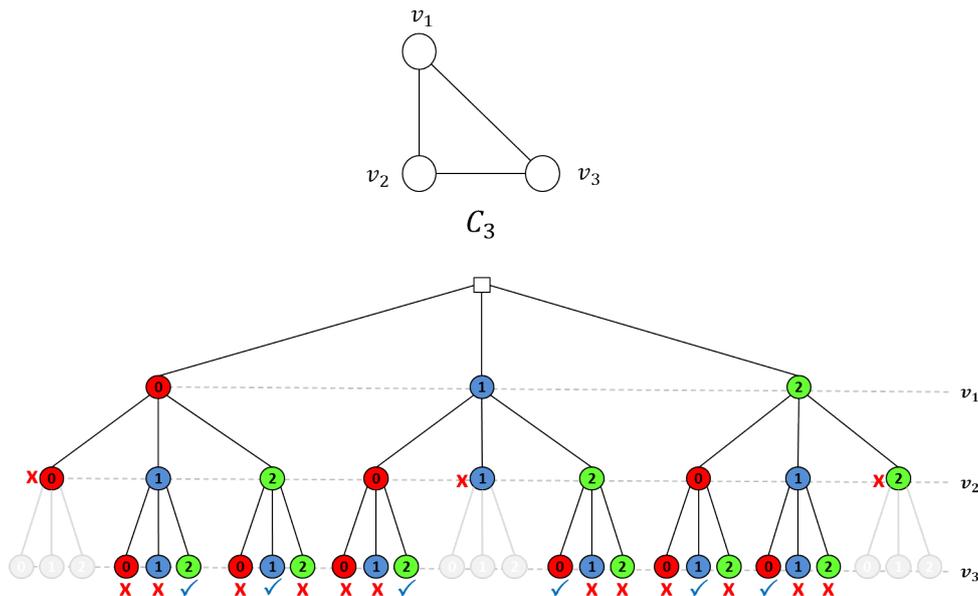


Figura 5.12: Exemplo de execução do algoritmo.

A melhor solução é aquela em que o grafo usa a menor quantidade de cores para sua coloração própria. O exemplo é um K_3 , logo a menor quantidade de cores utilizada é 3 (três). A Figura 5.13 apresenta todas as soluções encontradas na execução do Algoritmo 19. O algoritmo encontrou 6 soluções válidas.

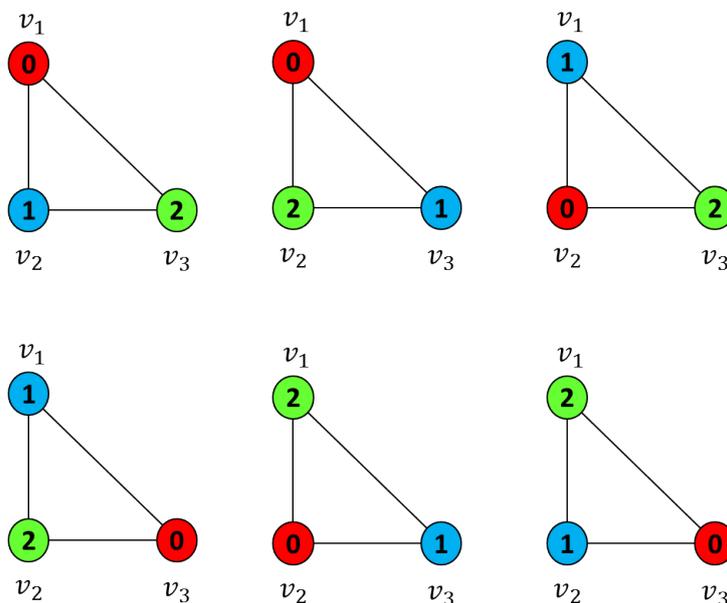


Figura 5.13: Soluções encontradas para o grafo da Figura 5.12

O algoritmo abordado é semelhante ao apresentado no Capítulo 4 para a k -selecionabilidade e também será abordado nesta seção, porém para determinar a k - (γ, μ) -

selecionabilidade. O algoritmo enumera **todas** as listas de tamanho k do tipo (γ, μ) para verificar se o grafo é colorível para todas essas listas de cores.

O algoritmo começa na Linha 1 com n recebendo o número de vértices. O algoritmo $\acute{e}_k\text{-}(\gamma, \mu)\text{-selecionável}$ é chamado dentro do *enquanto*, onde será realizado o cálculo as listas de cores do tipo γ, μ de tamanho k . Esse cálculo é feito pela fórmula $\{1, 2, \dots, k\}, \{2, 3, \dots, k + 1\}, \dots, \{n - k + 1, n - k + 2, n\}$. A quantidade de listas geradas é $n - (k - 1)$.

Formado os conjuntos de listas, o algoritmo forma os conjuntos de soluções de listas de cores, com elementos repetidos, ou seja, a permutação de listas de cores repetidas, dada pela função P_x , na linha 2. Para cada conjunto $L \in P$ de listas para o grafo, o algoritmo faz uma chamada a função *Existe_Lista_Coloracao* do Algoritmo 2 para verificar se o grafo possui coloração própria com este conjunto de listas.

Algoritmo 9: $k\text{-}(\gamma, \mu)\text{-selecionável}(G = (V, E))$

```

1  $n \leftarrow |V|$ 
2  $k \leftarrow 2$ 
3 enquanto  $!(\acute{e}_k\text{-}(\gamma, \mu)\text{-selecionável}(G, k))$  faça
4   |  $k \leftarrow k + 1$ 
5 fim enquanto
6 Retorne  $k$ 

```

Algoritmo 10: $\acute{e}_k\text{-}(\gamma, \mu)\text{-selecionável}(G = (V, E), k)$

```

1  $X \leftarrow \{1, 2, \dots, k\}, \{2, 3, \dots, k + 1\}, \dots, \{n - k + 1, n - k + 2, n\}$ 
2  $P_x \leftarrow$  todas as  $n$ -permutações de  $X$  com elementos repetidos
3 para cada  $L \in P_x$  faça
4   | se  $!(\text{Existe\_Lista\_Coloracao}(G, L, \text{nao}, 1))$  então
5     | Retorne NAO
6   | fim se
7 fim para cada
8 Retorne SIM

```

Se não for possível colorir, a função retorna *não* para a função $\acute{e}_k\text{-}(\gamma, \mu)\text{-selecionável}$, onde o algoritmo incrementa k e recomeça a montagem de novas listas de cores, agora

com $k + 1$. Se for possível colorir, a função retorna *sim* para a função $\acute{e}_k(\gamma, \mu)$ -selecionável, finalizando assim a execução do algoritmo.

Um exemplo de execução deste algoritmo pode ser verificado em um grafo C_4 . As listas tipo γ, μ são determinadas na Linha 1, e são $\{1, 2\}$, $\{2, 3\}$ e $\{3, 4\}$. A seguir, as soluções são formadas pelas listas de cores anteriores, tomados n a n , onde é possível elementos repetidos. Essa notação é dada por AR_p^n . O princípio multiplicativo diz que o número total de maneiras de se retirar n elemento dos p objetos, distintos ou não, é igual a

$$AR_p^n = p^n \quad (5.1)$$

uma vez que o primeiro elemento pode ser retirado de p maneiras, o segundo também de p maneiras, e assim sucessivamente, até que o n -ésimo seja escolhido.

Voltando ao caso da geração de soluções de listas de cores, o C_4 possui quatro vértices e três listas de cores de tamanho 2 (relembrando $\{1, 2\}$, $\{2, 3\}$ e $\{3, 4\}$). No caso, as listas de cores serão tomados n a n , podendo ter listas repetidas. Logo, o cálculo será feito de acordo com a fórmula da Equação 5.1 para calcular o número de todas as soluções de listas de cores. Considerando as variáveis da equação, n representa o número de vértices do grafo e p representa a quantidade de listas de cores geradas anteriormente:

$$p^n = 3^4 = 81 \quad (5.2)$$

Ou seja, para o grafo C_4 , com três possíveis listas de cores de tamanho 2, existe 81 soluções de lista coloração de tamanho k do tipo γ, μ .

A Figura 5.14 apresenta um gráfico comparando a quantidade de listas de cores geradas pelos algoritmos de k -selecionabilidade e k - (γ, μ) -selecionabilidade para dois grafos, um com quatro vértices e outro com cinco vértices. É possível perceber que a diferença na quantidade de listas de cores geradas é grande. A k - (γ, μ) -coloração gera menos que 50% da quantidade de listas geradas pelo algoritmo da k -selecionabilidade. A quantidade de listas de cores é determinada pelo valor de k .

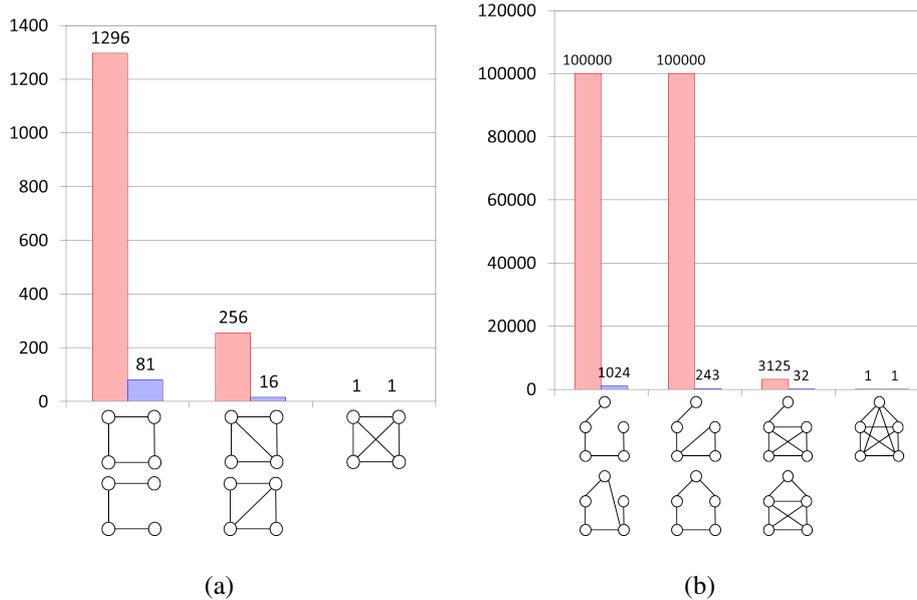


Figura 5.14: Gráficos comparando a quantidade de listas de cores gerados. As barras com a cor rosa indica a quantidade de listas geradas pelo algoritmo da k -selecionabilidade e as barras com a cor azul indica a quantidade de listas de geradas pelo algoritmo da k - (γ, μ) -selecionabilidade.

A Tabela 5.1 apresenta a quantidade de listas de cores geradas pelos dois algoritmos, de acordo com o valor de k e com a quantidade de vértices do grafo.

n/k	2		3		4	
	χ_ℓ	$\chi_{\ell,(\gamma,\mu)}$	χ_ℓ	$\chi_{\ell,(\gamma,\mu)}$	χ_ℓ	$\chi_{\ell,(\gamma,\mu)}$
2	1	1	-	-	-	-
3	27	8	1	1	-	-
4	1296	81	256	16	1	1
5	100000	1024	100000	243	3125	32
6	11390625	15625	64000000	4096	11390625	729
7	1801088541	279936	64339296875	78125	64339296875	16384
8	377801998336	5764801	96717311574016	1679616	576480100000000	390625
9	101559956668416	134217728	208215748530929664	40353607	8004512848309157376	10077696
10	34050628916015600	3486789401	640173642240000000000	1073741824	166798809782010000000000	282475249

Tabela 5.1: Quantidade de listas de cores geradas pelos algoritmos da k -selecionabilidade (χ_ℓ) e k - (γ, μ) -selecionabilidade ($\chi_{\ell,(\gamma,\mu)}$).

5.5 Considerações do capítulo

Este capítulo apresentou a principal contribuição deste trabalho. Foram apresentados o conceito da correlação entre (γ, μ) -coloração e selecionabilidade em grafos, a k - (γ, μ) -selecionabilidade. Além disso, foram apresentadas provas para duas classes de grafos, provando que grafos bipartidos completos são 2 - (γ, μ) -coloríveis e grafos periplanares, que são 3 - (γ, μ) -coloríveis, além de uma prova geral, de grafos que são k -coloríveis que são k - (γ, μ) -coloríveis. Foram apresentados algoritmos para essa coloração e o cálculo da quantidade de listas de cores geradas pelo algoritmo exato.

Capítulo 6

Considerações Finais

Nesta dissertação, foram apresentados o problema da lista coloração em grafos, suas variações μ -coloração e (γ, μ) -coloração, e as propriedades da selecionabilidade em grafos. Uma correlação entre a (γ, μ) -coloração com a propriedade de selecionabilidade foi proposta, gerando assim a k - (γ, μ) -coloração ou k - (γ, μ) -selecionabilidade. O problema da lista coloração e a selecionabilidade possuem muitas variações e propriedades ainda pouco exploradas, além do mais, a lista coloração também possui aplicação no problema prático de alocação de canais.

A revisão literatura foi elaborada com base na metodologia adotada pela revisão sistemática e apresentou os principais resultados na literatura de selecionabilidade em grafos, principalmente relacionados a diversos tipos de grafos, como grafos planares, grafos multipartidos e grafos proibidos. Destacam-se também os resultados que utilizam algumas das principais técnicas de provas de selecionabilidade em grafos. Alguns dos principais resultados em selecionabilidade e suas respectivas provas existentes na literatura foram apresentados com mais detalhes, além da prova detalhada dos teoremas de grafos $\theta_{2,2,2k}$ que são 2-selecionável e de um grafo que é 2-selecionável se, e somente se, o núcleo do grafo pertence a $T = \{K_1, C_{2k+2}, \theta_{2,2,2k}\}$, além do teorema de que todo grafo planar é 5-selecionável.

Com relação a k - (γ, μ) -coloração, algumas provas foram apresentadas utilizando as principais técnicas de provas em selecionabilidade. Foram provados que: grafos bipartidos completos são 2- (γ, μ) -selecionável, através da técnica do sistema dos representantes distintos e grafos periplanares são 3- (γ, μ) -selecionável através da técnica de degeneração em grafos. A prova de grafos que são k -coloríveis são k - (γ, μ) -selecionáveis comprovou

que algumas propriedades em selecionabilidade não são válidas em alguns casos para k - (γ, μ) -coloração. Um exemplo seria o grafo bipartido completo, que é 3-selecionável, porém é 2- (γ, μ) -selecionável.

Foram realizados estudos de algoritmos para lista coloração, selecionabilidade e k - (γ, μ) -coloração. Em lista coloração, apresentamos 4(quatro) algoritmos: um algoritmo exato, um algoritmo para verificar se existe coloração própria com as listas de cores dada, uma heurística clássica para lista coloração e um algoritmo para verificar se dada solução em lista coloração é válida. Além de um algoritmo para k -selecionabilidade e k - (γ, μ) -coloração, com base no teorema de que todo grafo tem uma k -coloração possui uma k - (γ, μ) -coloração.

Como atividades adicionais da pesquisa, no ano de 2012 ocorreu a participação no Congresso Latino-Iberoamericano Investigación Operativa/Simpósio Brasileiro de Pesquisa Operacional (CLAIO/SBPO) de 24 a 28 de setembro de 2012 no Rio de Janeiro, com apresentação de um resumo na forma de pôster envolvendo algoritmos apresentados, para a determinação do número lista cromático de uma coloração com restrições. Um artigo está em fase de submissão para a revista Matemática Contemporânea, como resultado da apresentação no evento Latin American Workshop on Cliques in Graphs (LAWCG) 2014.

Como trabalhos futuros, pretende-se estudar outros tipos de variações em lista coloração, se possível correlacionar a (γ, μ) -coloração com outras variações de coloração existentes na literatura, verificando assim, se as propriedades de uma dessas variações se mantêm, caso aja correlação dessa coloração com a (γ, μ) -coloração. Pretende-se ainda estudar os algoritmos, provas e/ou a complexidade computacional dessas variações, bem como verificar a existência de problemas em aberto.

Referências

- [1] AGEEV, A. A. Every circle graph of girth at least 5 is 3-colourable. *Discrete Mathematic* (1999), 229–233.
- [2] ALON, N. Restricted colorings of graphs. *Surveys in Combinatorics* (1993), 1–33.
- [3] BERTOSSI, A. L(2,1,1)-labeling problem on graph. *Discrete Applied Mathematics* (1999), 309–316.
- [4] BONDY, J. A., AND MURTY, U. S. R. *Graph Theory with Applications*. Elsevier Science Publishing, 1982.
- [5] BONOMO, F., AND MARENCO, J. Exploring the complexity boundary between coloring and list-coloring. *Annals of Operations Research* (2009), 3–16.
- [6] BORODIN, O. V. Structural properties of plane graphs without adjacent triangles and an application to 3-colorings. *Journal of Graph Theory* (1996), 183–186.
- [7] BRELAZ, D. New methods to color the vertices of a graph. *Communications of the ACM* (1979).
- [8] CHRISTOS H. PAPADIMITRIOU, K. S. *Combinatorial Optimization: Algorithms and Complexity*. Dover Books on Computer Science, 1998.
- [9] CHUDNOVSKY, M. Coloring graphs with forbidden induced subgraphs. *Mathematics Subject Classification* (2010).
- [10] CORMEN, T. H., LEISERSON E CHARLES E. AND RIVEST, R. L., AND STEIN, C. *Algoritmos - Teoria e Prática*, 2th ed. Editora Campus, 2002.
- [11] DA FAZENDA, M. *Receitas Públicas - Manual de Procedimentos - Aplicado à União, Estados, Distrito Federal e Municípios*. 2004.

- [12] DE RIDDER, H. Information system on graph classes and their inclusions, 2015.
- [13] DE SOUZA, S. D. Partições em grafos : Caracterizações, algoritmos e complexidade. *Tese de Doutorado* (2002).
- [14] E. A. NORDHAUS, J. W. G. On complementary graphs. *The American Mathematical Monthly* (1956), 175–177.
- [15] E BAOYINDURENG WU, L. Z. A note on 3-choosability of planar graphs without certain cycles. *Discrete Mathematics* (2005).
- [16] FAN WANG, W. Planar graphs that have no short cycles with a chord are 3-choosable. *Taiwanese Journal of Mathematics* (2007), 179–186.
- [17] FAN WANG E KO-WEI LIH, W. Choosability and edge choosability of planar graphs without intersecting triangles. *Journal on Discrete Mathematics* (2002), 538–545.
- [18] FINCK, H. J. On the chromatic number of a graph and its complements. *Theory of Graphs*.
- [19] FLAVIA BONOMO, M. C. Between coloring and list-coloring: μ -coloring. *Electronic Notes in Discrete Mathematics* 19 (2005), 117–123.
- [20] FLEISCHNER, H., AND STIEBITZ, M. A solution to a colouring problem of paul erdos. *Discrete Math* (1992), 39 – 48.
- [21] FLÁVIA BONOMO, G. D. Graph coloring problems. *XII ELAVIO, February 2007, Itaipava, Brasil* (2007).
- [22] FLÁVIA BONOMO, YURI FAENZA, G. O. On coloring problems with local constraints. *ELetronic Notes in Discrete Mathematics* (2009), 215–220.
- [23] FREDERIC MAFFRAY, S. G. Graphs whose choice number is equal to their chromatic number. *John Wiley and Sons* (1998).
- [24] GARY CHARTRAND, P. Z. *Chromatic Graph Theory*. CRC Press, 2008.
- [25] GOLUMBIC, M. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, 1980.

- [26] GUIZHEN LIU, J. H. E. J. W. On 3-degeneracy of some c_7 -free plane graphs with application to choosability. *Ars Combinatoria* (2009).
- [27] HALE, W. K. Frequency assignment: Theory and applications. *Proceedings of the IEEE* (1980).
- [28] HALL, M. Distinct representatives of subsets. *Bulletin of the Institute of Combinatorics and its Applications* (1948).
- [29] HENRY. A. KIERSTEAD, ANDREW SALMON, R. W. On the choice number of complete multipartite graphs with part size four. *Mathematics Subject Classification* (2014).
- [30] HIKOE ENOMOTO, KYOJI OHBA, K. O. J. S. Choice number of some complete multi-partite graphs. *Discrete Mathematics* (2002), 55–66.
- [31] HUDSON V. KRONK, A. T. W. A 4-color theorem for toroidal graphs. *Proceedings of the American Mathematical Society* (1972), 83–86.
- [32] JAN KRATOCHVIL, Z. T. Algorithmic complexity of list colorings. *Discrete Applied Mathematics* (1994), 297 – 302.
- [33] JANSEN, K. The optimum cost chromatic partition problem. *Lecture Notes in Computer Science* (1997), 25–36.
- [34] K. JANSEN, P. S. Generalized colorings for tree-like graphs. *Discrete Applied Mathematics* (1997), 135–155.
- [35] KARP, R. M. Reducibility among combinatorial problems. *Complexity of Computer Computations* (1972), 85–103.
- [36] KENNETH I. APPEL, W. H. Every planar map is four colourable, part *i*. *Illinois Journal of Mathematics* (1995), 429–567.
- [37] KIERSTEAD, H. A. On the choosability of complete multipartite graphs with part size three. *Discrete Mathematics* (2000), 255–259.
- [38] KLAUS JANSEN, P. S. Generalized coloring for tree-like graphs. *Discrete Applied Mathematics* (1997), 135–155.

- [39] KÖNIG, D. Über graphen und ihre anwendung auf determinantentheorie und mengenlehre. *Mathematische Annalen* (1916), 453 – 465.
- [40] KUBALE, M. Some results concerning the complexity of restricted colorings of graphs. *Discrete Applied Mathematics* (1992).
- [41] LE, V. B., BRANDSTÄDT, AND SPINRAD, J. *Graph Classes: A Survey*. 1999.
- [42] LI, X. On 3-choosable planar graphs of girth at least 4. *Discrete Mathematics* (2009), 2424–2431.
- [43] LI ZHANG, B. W. A note on 3-choosability of planar graphs without certain cycles. *Discrete Mathematics* (2005), 206–209.
- [44] LIANG SHEN, Y. W. A sufficient condition for a planar graph to be 3-choosable. *Information Processing Letters* (2007), 146–151.
- [45] LICK DON, W. A. k -degenerate graphs. *Canadian Journal of Mathematics* (1970), 1082–1096.
- [46] LIDICKÝ, B. On 3-choosability of plane graphs having no 3-, 6-, 7- and 8-cycles. *Australasian Journal of Combinatorics* (2009), 77–86.
- [47] LOUIS ESPERET, A. G. E. F. M. List-coloring claw-free graphs with small clique number. *Graphs and Combinatorics* (2014), 365–375.
- [48] LU ENYUE, Z. K. The list chromatic numbers of some planar graphs. *Applied Mathematics* (1998), 108–116.
- [49] MARCO GOLDBARG, HENRIQUE PACCA, L. L. *Otimização Combinatória e Programação Linear*. Editora Campus, 2005.
- [50] MARTIN AIGNER, G. M. Z. *The Algorithm Design Manual, 2nd Edition*. 2008.
- [51] MARTIN GROTSSCHEL, LÁSZLÓ LOVÁSZ, A. S. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* (1981), 169–197.
- [52] MCKAY, B. A note on the history of the four-colour conjecture. *Research School of Computer Science* (2012), 1–2.

- [53] MIRZAKHANI, M. A small non-4-choosable planar graph. *Bulletin of the Institute of Combinatorics and its Applications* (1996), 15–18.
- [54] MONTASSIER, M. A note on the not 3-choosability of some families of planar graphs. *Information Processing Letters* (2006), 68–71.
- [55] MOSHE DROR, GERD FINKE, S. G. W. K. On the complexity of a restricted list-coloring problem. *Discrete Mathematics* (1999), 103–109.
- [56] NEIL ROBERTSON, DANIEL P. SANDERS, P. S. E. R. T. A new proof of the four colour theorem. *Electronic Research Announcements of the American Mathematical Society* (1996).
- [57] NOGA ALON, M. T. Colorings and orientations of graphs. *Combinatoria* (1992), 125–134.
- [58] OHBA, K. The list chromatic index of a bipartite multigraph. *Jornal Combinatoria Theory* (1995).
- [59] PAUL ERDOS, ARTHUR RUBIN, H. T. Chosability in graphs. *Proceedings, West Coast Conference on Combinatorics* (1979).
- [60] PETER C.B. LAM, WAI CHEE SHIU, Z. M. S. The 3-choosability of plane graphs of girth 4. *Discrete Mathematics* (2005), 297–301.
- [61] PETER CHE BOR LAM, BAOGANG XU, J. L. The 4-choosability of plane graphs without 4-cycles. *Journal of Combinatorial Theory* (1999), 117–126.
- [62] RISTE SKREKOVSKI, ZDENEK DVORAK, B. L. Graphs with two crossings are 5-choosable. *Discrete Mathematics* (1998), 1746–1453.
- [63] SKIENA, S. *Proofs from the book. The Dinitz problem.* 1998.
- [64] SKREKOVSKI, R. Choosability of k_5 -minor-free graphs. *Discrete Mathematics* (1998), 223–226.
- [65] SYLVAIN GRAVIER, F. M. Choice number of 3-colorable elementary graphs. *Discrete Mathematics* (1997), 353–?358.

- [66] SYLVAIN GRAVIER, F. M. Graphs whose choice number is equal to their chromatic number. *Journal of Graph Theory* (1997), 87–97.
- [67] SZWARCFITER, J. L. *Grafos e algoritmos computacionais*. Campus.
- [68] SZWARCFITER, J. L. *Grafos e Algoritmos Computacionais*. Elsevier, 1986.
- [69] THOMASSEN, C. Every planar graph is 5-choosable. *Journal of Combinatorial Theory* (1994), 180–181.
- [70] THOMASSEN, C. 3-list-coloring planar graphs of girth 5. *Journal of Combinatorial Theory* (1995), 101–107.
- [71] THOMASSEN, C. The chromatic number of a graph of girth 5 on a fixed surface. *Journal of Combinatorial Theory* (2003), 38–71.
- [72] TOMMY R. JENSEN, B. T. *Graph Coloring Problems*. Wiley-Interscience, New York, 1995.
- [73] VIZING, V. G. Vertex coloring of a graph with assigned colors. *Metody Diskret. Analiz. (Novosibirsk)* (1976), 3–10.
- [74] VOIGT, M. List colourings of planar graphs. *Discrete Mathematics* (1993), 215–219.
- [75] VOIGT, M. A not 3-choosable planar graph without 3-cycles. *Discrete Mathematics* (1995), 325–325.
- [76] VOIGT, M. On list colourings and choosability of graphs.
- [77] VOIGT, M. A non-3-choosable planar graph without cycles of length 4 and 5. *Discrete Mathematics* (2007).
- [78] WEI-FAN WANG, K.-W. L. Choosability, edge choosability e total choosability of outerplane graphs. *European Journal of Combinatorics* (2001), 71–78.
- [79] WEI-FAN WANG, K.-W. L. Choosability, edge choosability of planar graphs without five cycles. *Applied Mathematics Letters* (2002), 561–565.
- [80] WOODALL, D. R. List colourings of graphs. *Surveys in Combinatorics* (2001), 269–301.

- [81] XIAOFANG, L. The 4-choosability of toroidal graphs without intersecting triangles. *Proceedings of the American Mathematical Society* (2007), 85 – 91.
- [82] YINGQIAN WANG, HUAJING LU, M. C. A note on 3-choosability of planar graphs. *Information Processing Letters* (2007), 206–211.
- [83] YINGQIAN WANG, HUAJING LU, M. C. Planar graphs without cycles of length 4, 5, 8, or 9 are 3-choosable. *Discrete Mathematics* (2010), 147 – 158.
- [84] YINGQIAN WANG, Q. Z. On 3-choosability of triangle-free plane graphs. *Science China Mathematics* (2011), 1287–1298.
- [85] ZDENEK DVORAK, BERNARD LIDICKÝ, R. S. Planar graphs without 3-, 7-, and 8-cycles are 3-choosable. *Discrete Mathematics* (2009), 5899 – 5904.
- [86] ZDENEK DVORAK, BERNARD LIDICKÝ, R. S. 3-choosability of triangle-free planar graphs with constraint on 4-cycles. *Discrete Mathematics* (2010), 934?–945.
- [87] ZHANG, H. On 3-choosability of planar graphs with neither adjacent triangles nor 5-, 6- and 9-cycles. *Information Processing Letters* (2010), 354–356.