

UNIVERSIDADE FEDERAL FLUMINENSE

MATHEUS SOUZA D'ANDREA ALVES

COLORAÇÃO DE GRAFOS(r, ℓ)

Niterói

2018

UNIVERSIDADE FEDERAL FLUMINENSE

MATHEUS SOUZA D'ANDREA ALVES

COLORAÇÃO DE GRAFOS(r, ℓ)

Trabalho de Conclusão de Curso apresentado
à Universidade Federal Fluminense como re-
quisito parcial para a obtenção do Grau de
Bacharel em Ciência da Computação.

Orientador:

Dr. Uéverton dos Santos Souza

Niterói

2018

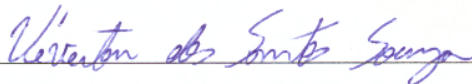
MATHEUS SOUZA D'ANDREA ALVES

Coloração de $Grafos(r, \ell)$

Trabalho de Conclusão de Curso apresentado
à Universidade Federal Fluminense como re-
quisito parcial para a obtenção do Grau de
Bacharel em Ciência da Computação.

Aprovada em 04/07/2018.

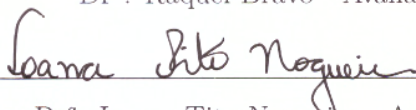
BANCA EXAMINADORA



Dr. Uéverton dos Santos Souza - Orientador, UFF
(Presidente)



Dr^a. Raquel Bravo - Avaliadora, UFF



Dr^a. Loana Tito Nogueira - Avaliadora, UFF

Niterói

2018

Resumo

Um problema clássico na literatura é o problema de coloração própria de um grafo, isto é, encontrar uma q – *coloração* para um grafo G tal que todo vértice $v \in V(G)$ não possua nenhum vizinho da mesma cor e q seja mínimo. Esse problema é conhecido ser NP-Difícil para grafos gerais. O trabalho a seguir tem como proposta desvendar e catalogar a complexidade clássica e parametrizada de tal problema para a classe de Grafos(r, ℓ), i.e. grafos particionáveis em r conjuntos independentes e ℓ cliques; Identificando as características que tornam o problema difícil e a relação do problema de coloração com outros problemas, quando abordado pela perspectiva parametrizada.

Palavras-chave: Complexidade parametrizada. Grafos(r, ℓ). Partição de grafos. Coloração de Grafos

Abstract

A classical problem in graph theory is the problem of proper coloring a graph, i.e. to find a q – *coloring* for a graph G such that every vertex $v \in V(G)$ does not have any neighbor of the same color and q is the smallest possible number, a problem known to be NP-Hard for a general graphs. The following work attempts to uncover and catalog the parametrized complexity of such problem for the class of graphs(r, ℓ), i.e. partitionable graphs in r independent sets and ℓ cliques; Identifying the characteristics that make the problem hard and the relation of the stated problem to other problems when approached by the parameterized perspective.

Keywords: Parametrized Complexity. Graph(r, ℓ). Graph Partitioning. Graph Coloring.

Lista de Figuras

2.1	Grafo G : Transformação de 3-SAT em co-bipartido com foco na cláusula P	19
3.1	Grafo G' inicial.	25
3.2	Grafo G' final.	27
3.3	Gadget com vértices de lista um reproduzindo vértice de lista um em vértice de lista três	27
3.4	Demonstração de coloração para vizinhança de tamanho um.	29
3.5	Demonstração de coloração para vizinhança de tamanho dois com cores compartilhadas.	30
3.6	Estrutura Γ e suas possíveis colorações.	31

Lista de Tabelas

2.1	1ª Dicotomia P/NP_c parcial do problema de coloração em $\text{Grafos}(r, \ell)$. . .	14
2.2	2ª Dicotomia P/NP_c parcial do problema de coloração em $\text{Grafos}(r, \ell)$. . .	16
2.3	Dicotomia P/NP_c do problema de coloração em $\text{Grafos}(r, \ell)$	20
4.1	Dicotomia P/NP_c do problema de clique cover em $\text{Grafos}(r, \ell)$	33

Conteúdo

1	Introdução	9
1.1	Organização do trabalho	10
1.2	Estruturas básicas	10
1.3	Problemas abordados	11
1.4	Complexidade computacional	11
1.4.1	NP-Completeness	12
1.5	Complexidade parametrizada	12
1.5.1	Tratabilidade Parametrizada	12
1.5.2	Intratabilidade Parametrizada	12
2	Análise clássica para coloração em Grafos(r, ℓ)	14
3	Análise parametrizada para coloração em Grafos(2,1)	21
3.1	O tamanho do menor conjunto independente como parâmetro	21
3.2	O tamanho do maior independente como parâmetro	23
3.3	O tamanho da clique como parâmetro	23
3.4	A vizinhança da clique como parâmetro	24
3.4.1	Apenas vértices com listas de tamanho um ou três	25
3.4.2	Apenas vértices com listas de tamanho dois ou três	28
3.5	Tamanho da não vizinhança da clique como parâmetro	32
4	Conclusão	33
4.1	Resultados e consequências	33

4.2	Trabalhos futuros	34
-----	-----------------------------	----

Referências	35
--------------------	-----------

Capítulo 1

Introdução

Uma das principais motivações do estudo de classes de grafos é o fato de que diversos problemas, que são difíceis para grafos em geral, tornam-se tratáveis quando restritos a classes especiais de grafos. Assim, busca-se delimitar a partir de que ponto um determinado problema pode ser resolvido de forma eficiente. Particularmente, o problema de particionamento em grafos tem despertado muito interesse devido às pesquisas de grafos perfeito e também pela procura de algoritmos eficientes para o reconhecimento de determinadas classes de grafos. O problema de partição de grafos pode ser descrito como tendo por objetivo particionar o conjunto dos vértices de um grafo em subconjuntos V_1, V_2, \dots, V_k onde $V_1 \cup V_2 \cup \dots \cup V_k = V$ e $V_i \cap V_j = \emptyset$, $i \neq j$, $1 \leq i \leq k$ e $1 \leq j \leq k$, exigindo-se, porém, algumas propriedades sobre estes subconjuntos de vértices. Estas propriedades podem ser *internas*, como por exemplo exigir que os vértices de cada subconjunto V_i sejam dois-a-dois adjacentes (isto é, V_i seja uma clique) ou dois-a-dois não-adjacentes (isto é, V_i seja um conjunto independente), ou *externas*, onde as exigências são feitas sobre os pares de $V_i \times V_j$, podem ser adjacentes ou não-adjacentes entre si.

Um problema bem conhecido é o de verificar se um dado grafo G é split, isto é, verificar se os vértices de G podem ser particionados em dois subconjuntos, dos quais um é independente e o outro uma clique. Brandstädt foi o responsável por generalizar a definição de split no que chamamos aqui de Grafos(r, ℓ), um grafo que pode ser particionado em r conjuntos independentes e ℓ cliques.

Em outra perspectiva o problema de coloração de vértices de um grafo também é de grande interesse, tendo suas aplicações em diversas áreas; *Pattern matching*, escalonamento em esportes e no trânsito, e resolução de problemas de *Sudoku* são alguns exemplos de problemas onde a coloração de grafos pode ser empregada[11]. O problema de coloração mínima dos vértices de um grafo chamado também de coloração própria de um grafo

é descrito como a atribuição de cores em um grafo G de forma que seja possível atribuir a cada um dos vértices de G uma entre k cores de forma que, dado quaisquer dois vértices vizinhos em G eles não compartilhem uma mesma cor e k seja o menor número onde tal restrição é atingida.

1.1 Organização do trabalho

O intuito deste trabalho é o de desvendar e classificar a dificuldade do problema de coloração em $\text{Grafos}(r, \ell)$. Ao nos aprofundar na investigação mostraremos ainda a proficuidade dos tamanhos das partições para a extração de um algoritmo FPT.

O capítulo sobre análise computacional clássica se propõe a demonstrar uma dicotomia *Polinomial / NP-Completo* para o problema abordando a quantidade de partições. Uma vez definida tal dicotomia o capítulo de análise parametrizada busca através de parâmetros encontrar algoritmos tratáveis por parâmetro fixo para solucionar o problema em questão. Concluiremos o escrito resumizando nossas descobertas e esclarecendo as repercussões de nosso trabalho e os novos desafios que nos pode trazer.

As seções mostradas a seguir se propõem a esclarecer as estruturas, problemas e ferramental utilizado durante o trabalho.

1.2 Estruturas básicas

Um Grafo G é uma estrutura que contém um conjunto de vértices $V(G) = \{v_1, v_2, \dots, v_n\}$ e um conjunto de arestas $E(G) = \{(v, j) | \{v, j\} \subset V(G)\}$, dizemos que o vértice v pertence ao grafo G se $v \in V(G)$, e que existe uma aresta entre v e u se $(u, v) \in E(G)$. Nesse trabalho usaremos apenas grafos não direcionados, dessa forma $(u, v) \equiv (v, u)$.

Um Grafo dito $\text{Grafo}(r, \ell)$ ou abreviadamente $G(r, \ell)$ é qualquer grafo pertencente à classe dos grafos que podem ser particionados em r conjuntos independentes ℓ cliques. Considere aqui que assumiremos que as partições dadas em um $\text{Grafo}(r, \ell)$ podem estar vazias, isto é, não possuem nenhum vértice.

Um conjunto independente $R \subseteq G$ é uma partição de G tal que: $\{v, u\} \subset V(R) \implies \nexists (v, u) \in E(R)$. Já uma clique $L \subseteq G$ é uma partição de G tal que: $\{v, u\} \subset V(L) \implies \exists (v, u) \in E(L)$. Dizemos que um grafo G possui uma bipartição quando todos os seus vértices podem ser divididos em dois conjuntos independentes disjuntos. Um conjunto

independente ou clique é dita máximo ou máxima respectivamente, se tal partição contém o maior número possível de vértices que mantém sua propriedade.

Um grafo é completo se todos os seus vértices são adjacentes entre si, e nulo se nenhum vértice é adjacente a outro.

Denominamos por $G[S]$ o grafo induzido por S em G onde $S \subset V(G)$.

1.3 Problemas abordados

Definição 1. *Coloração própria de Grafos.*

Entrada: *Um grafo G e um inteiro k .*

Pergunta: *É possível atribuir a cada vértice pertencente à G uma entre k cores de tal forma que dado quaisquer dois vértices adjacentes eles tenham cores distintas e k seja o mínimo de cores possível?*

É importante notarmos que, por motivos de simplicidade sempre que nos referirmos ao problema de Coloração de um grafo, nos referiremos à Coloração Própria de Grafos, a menos que dito o contrário. Observe também que k é conhecido como o número cromático de G se G possui uma k -coloração própria.

Definição 2. *Lista coloração de Grafos.*

Entrada: *Uma paleta de cores P e um Grafo G onde todo $v \in V(G)$ pode ser colorido com um subconjunto $S_v \subset P$.*

Pergunta: *É possível escolher uma cor dentro das de S_v para todo vértice v de forma que dado quaisquer dois vértices adjacentes eles tenham cores distintas?*

Note que quando $|S_v| = |V(G)|$ o problema é o Problema de Coloração clássico.

1.4 Complexidade computacional

Um algoritmo de tempo polinomial é um algoritmo cuja função de complexidade de tempo é $\mathcal{O}(p(n))$, para alguma função polinomial p , onde n é usado para denotar o tamanho da entrada.

Um problema Π pertence à classe P se e somente se Π pode ser solucionado em tempo polinomial por algum algoritmo determinístico. Um problema Π pertence à classe NP se e somente se para um dado certificado afirmativo há um algoritmo que verifica sua validade em tempo polinomial.

Dados dois problemas Π e Π' dizemos que $\Pi \propto \Pi'$ (Π se reduz à Π' em tempo polinomial) se existe um algoritmo capaz de construir uma instância J de Π' a partir de uma instância I de Π em tempo polinomial, tal que a partir de uma resposta para J se tenha uma resposta para I em tempo polinomial.

1.4.1 NP-Compleitude

Um problema Π' é dito *NP – Difícil* se todo problema $\Pi \in NP$ se reduz à Π' . Se $\Pi' \in NP$ então Π' é *NP – Completo*.

1.5 Complexidade parametrizada

1.5.1 Tratabilidade Parametrizada

Definição 3. Dado um problema Π e um conjunto de aspectos de Π chamado $S = \{s_1, s_2, s_3, \dots, s_n\}$ denotamos por $\Pi(S)$ o problema Π parametrizado por S .

Definição 4. Dado um problema parametrizado $\Pi(S)$ dizemos que o mesmo é *FPT* (Fixed parameter tractable (Tratável por parâmetro fixo)) se existe um algoritmo capaz de resolver Π em $\mathcal{O}(f(S) \times n^c)$ onde $f(S)$ é uma função arbitrária e c uma função $\mathcal{O}(1)$.

1.5.2 Intratabilidade Parametrizada

Esta seção irá sumarizar as definições de *W – hierarquia* estabelecida por Downey et al.[4], para tanto observe as seguintes definições.

Definição 5. Sejam $\Pi(k)$ e $\Pi'(k')$ onde $k' \leq g(k)$. Chamamos de *FPT-redução* de $\Pi(k)$ para $\Pi'(k')$ é uma transformação R quando:

- $\forall x, x \in \Pi(k) \iff R(k) \in \Pi'(k')$
- R é computável por um *FPT-Algoritmo*, com relação a k

Definição 6. *Satisfabilidade Ponderada em circuitos de entrelaçamento t e profundidade h $WCS(t, h)$.*

Entrada: *um circuito de decisão C com entrelaçamento t e profundidade h .*

Pergunta: *C possui uma atribuição satisfatível?*

Usando as definições 5 e 6 definiremos então a pertinência de um problema à classe $W[t]$.

Definição 7. *Um problema parametrizado $\Pi(k)$ pertence a classe $W[t]$ se e somente se existe uma FPT-Redução de tal problema para $WCS(t, h)$ para algum h constante. Logo devido a transitividade de FPT-Redução, se existe uma FPT-Redução de qualquer problema $\Pi'(k')$ para $\Pi(k)$ então $\Pi(k) \in W[t]$*

Capítulo 2

Análise clássica para coloração em Grafos(r, ℓ)

O problema de coloração aplicado a Grafos(r, ℓ) pode ser facilmente solucionável em alguns casos, por exemplo, um grafo sem arestas i.e um Grafo(1,0) é colorível com apenas uma cor, um grafo completo, ou seja um Grafo(0,1), é colorível com n cores onde n é a quantidade de vértices desse grafo, um grafo split, por sua vez, é um Grafo(1,1) e pode ser colorido utilizando k cores, onde k é o tamanho da clique máxima do grafo, pois assumindo uma partição (1,1) onde a clique é máxima, cada vértice do conjunto independente pode ser colorível com alguma cor já presente na clique. E por fim, grafos bipartidos são coloridos com 2 cores uma cor para cada conjunto independente.

Sendo assim, sabemos que coloração é um problema solucionável em tempo polinomial em grafos completos, nulos, splits e bipartidos. Portanto, temos como ponto de partida para a nossa análise as questões apresentadas na Tabela 2.1.

$r \backslash \ell$	0	1	2	3	4	...	n
0	P	P	?	?	?	...	?
1	P	P	?	?	?	...	?
2	P	?	?	?	?	...	?
3	?	?	?	?	?	...	?
4	?	?	?	?	?	...	?
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	?
n	?	?	?	?	?	...	?

Tabela 2.1: 1ª Dicotomia P/NP parcial do problema de coloração em Grafos(r, ℓ)

Neste ponto, inciaremos a nossa discussão sobre a complexidade do problema de co-

loração para as demais classes de grafos (r, ℓ).

Teorema 1. *Coloração de Grafo(0,2) pode ser solucionado em tempo polinomial.*

Demonstração. Um Grafo(0,2) G , também conhecido como grafo co-bipartido, é um grafo onde seu conjunto de vértices pode ser particionado em dois conjuntos K_1 e K_2 tal que $V(G) = K_1 \cup K_2$, $K_1 \cap K_2 = \emptyset$ e $G[K_1]$, $G[K_2]$ são grafos completos. É bem conhecido na literatura que um grafo co-bipartido é perfeito [2], consequentemente, seu número cromático é igual ao tamanho da maior clique. Note que encontrar a clique máxima de um grafo co-bipartido é equivalente a encontrar o conjunto independente máximo do complemento de G que, por sua vez, pode ser feito em tempo polinomial através de um algoritmo para Cobertura por Vértices em grafos bipartidos [10, 3]. Portanto, ao encontrar a cobertura por vértices mínima para o complemento de G encontramos a clique máxima de G , e como G é um grafo perfeito, sabemos resolver o problema de coloração em tempo polinomial. \square

A seguir apresentamos outra classe em que o problema pode ser solucionado em tempo polinomial.

Teorema 2. *Coloração de Grafo(3,0) é Polinomial.*

Demonstração. Se G é um grafo que admite um partição (3,0), sabe-se que o número cromático de G é no máximo 3, sendo assim, para determinarmos o valor exato de seu número cromático, basta verificar se G é um grafo vazio (número cromático igual a um), e caso contrário verificar se G é bipartido (número cromático igual a dois), como ambas verificações são executáveis em tempo polinomial podemos afirmar que o número cromático de um grafo(3,0) pode ser determinado em tempo polinomial. \square

Como o número cromático de grafos(3,0) pode ser solucionável em tempo polinomial, uma pergunta interessante passa a ser a complexidade de determinar o número cromático de um grafo (4,0).

Teorema 3. *Coloração de Grafo(4,0) é NP-Completo.*

Demonstração. Sabemos que todo grafo planar é 4-colorível [1], logo os grafos planares formam uma subclasse dos grafos(4,0). Portanto, para determinar o número cromático de um grafo planar G é necessário solucionar 3-coloração em grafos planares, no entanto, tal problema é NP-Completo[???], logo descobrir o número cromático de um grafo planar G é NP-Completo e consequentemente resolver coloração em grafos(4,0) é NP-Completo. \square

É importante notar que, todo $Grafo(r, \ell)$ é simultaneamente um $Grafo(r, \ell + 1)$ e um $Grafo(r + 1, \ell)$ já que por definição algumas partes podem ser vazias, portanto se o problema de coloração é NP-Completo para $Grafo(r, \ell)$ então ele é NP-Completo para $Grafo(r + 1, \ell)$ e $Grafo(r, \ell + 1)$.

Os resultados anteriores e a monotonicidade observada anteriormente no leva ao seguinte cenário:

$r \backslash \ell$	0	1	2	3	4	...	n
0	P	P	P	?	?	...	?
1	P	P	?	?	?	...	?
2	P	?	?	?	?	...	?
3	P	?	?	?	?	...	?
4	NPc	NPc	NPc	NPc	NPc	...	NPc
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	NPc
n	NPc	NPc	NPc	NPc	NPc	...	NPc

Tabela 2.2: 2ª Dicotomia P/NPc parcial do problema de coloração em Grafos(r, ℓ)

Como podemos observar, ainda nos falta mostrar a complexidade para alguns casos de fronteira, que necessitam de uma demonstração mais complexa.

Nesto ponto, iremos apresentar uma ferramenta importante que nos auxiliará nos demais resultados.

Teorema 4. *O problema de lista coloração de um grafo (r, ℓ) , se reduz ao problema de coloração de um grafo $(r, \ell + 1)$.*

Demonstração. Seja G uma instância do problema de lista coloração em grafos(r, ℓ), isto é, cada vértice $v \in V(G)$ possui uma lista de cores S_v . Seja $C = \{c_1, c_2, \dots, c_k\}$ a paleta de cores utilizada em G . A partir de G criaremos um grafo H_G , instância do problema de coloração, da seguinte forma:

- inicialmente faça $V(H_G) = V(G)$ e $E(H_G) = E(G)$;
- adicione em H_G uma clique K formada por k novos vértices, w_1, w_2, \dots, w_k , cada vértice $w_i \in V(K)$ representará a cor c_i presente em C ;
- para todo vértice $v \in V(H_G) \cap V(G)$ e todo vértice $w_i \in V(K)$ adicione uma aresta (u_i, v) em H_G se v não possui a cor c_i na lista coloração S_v em G .

Neste ponto nos resta demonstrar que G possui uma lista coloração própria se e somente se H_G é k -colorível, onde para k é o número de cores da paleta C de G .

Suponha que G possui uma lista coloração. Iremos mostrar que H_G admite um k -coloração como segue. Atribua para cada vértice em $V(H_G) \cap V(G)$ a mesma cor que lhe foi atribuída em G . Note que a clique K possui exatamente k vértices, consequentemente para colorirmos K precisaremos de k cores, assumimos que w_1 será colorido com c_1 , w_2 será colorido com c_2 e assim por diante. Por construção, só existe aresta de w_i para um vértice $v \in V(H_G) \setminus V(K)$ se v não possui c_i em sua lista de cores, portanto a coloração atribuída a clique K não conflita com as cores atribuídas aos vértices em $V(H_G) \cap V(G)$, como a paleta de G possui k cores, temos que H_G é k -colorível.

Suponha que o grafo H_G possua uma k -coloração. Por construção, k é a cardinalidade de K , observe que a remoção de K não afeta a coloração de $H_G \setminus K$. Como H_G é k -colorível e a clique K possui k vértices todas as cores de tal k -coloração estão presentes em K . Sem perda de generalidade, podemos assumir que as cores c_1, c_2, \dots, c_k estão atribuídas aos vértices w_1, w_2, \dots, w_k , respectivamente. Por construção de H_G todo par v, w_i onde $v \in V(H_G) \setminus V(K)$ e $w_i \in V(K)$ é não adjacente se e somente se o vértice v não possui c_i em sua lista coloração no grafo G , logo a coloração atribuídas aos vértices em $H_G \setminus K$ formam uma solução para lista coloração em G . Portanto G é uma instância sim de lista coloração. \square

A partir do Teorema 4, obtemos os seguintes corolários:

Corolário 1. *O problema de coloração é NP-Completo para Grafos(1, 2).*

Demonstração. Segue da NP-Completude de lista coloração em Grafos(1, 1), demonstrado por Jansen et al.[8]. \square

Corolário 2. *O problema de coloração é NP-Completo em Grafos(2, 1).*

Demonstração. Segue da NP-Completude de lista coloração em grafos bipartido demonstrado por Fellows et al.[5]. \square

Neste ponto, nos resta determinar a complexidade coloração em grafos (0,2).

Teorema 5. *Lista coloração é NP-Completo para Grafos(0,2).*

Demonstração. Para essa demonstração nos basearemos em um resultado obtido por Jansen [7]. A demonstração se baseia em realizar uma redução do problema 3-SAT restrito

para lista coloração de co-bipartido i.e. Grafo(0,2). Suponha o problema 3-SAT com as seguintes restrições:

- cada cláusula c_i contém dois ou três terminais.
- cada literal ou sua negação aparece no máximo em 3 cláusulas

Construiremos agora uma instância de lista coloração da seguinte forma:

Para cada literal j crie seis vértices: $a_j^{(1)}, a_j^{(2)}, a_j^{(3)}; b_j^{(1)}, b_j^{(2)}, b_j^{(3)}$. Atribuindo a cada vértice uma lista de cores da seguinte forma:

$$a_j^{(k)} \leq \{x_j^{(k)}, \overline{x_j^{(k)}}\}; b_j^{(k)} \leq \{\overline{x_j^{(k)}}, x_j^{((k \pmod{3})+1)}\}$$

Definimos como A o conjunto de todos os $a_j^{(k)}$ e B o conjunto de todos os $b_j^{(k)}$ e construímos uma clique com os vértices de A e B. Observe que só existem duas maneiras de se colorir este grafo:

- (1) $f(a_j^{(k)}) = x_j^{(k)} \Rightarrow b_j^{(k)} = \overline{x_j^{(k)}}$
- (2) $f(a_j^{(k)}) = \overline{x_j^{(k)}} \Rightarrow b_j^{(k)} = x_j^{((k \pmod{3})+1)}$

Agora, para cada cláusula definimos um vértice c_i e sua lista de cores da seguinte forma: para cada literal j ou sua negação \bar{j} presente na cláusula adicionamos à lista de c_i o $x_j^{(k)}$ onde k é o índice de ocorrência do literal ou de sua negação.

Por exemplo, suponha o seguinte 3-SAT:

$$(p \vee q \vee r) \wedge (\neg p \vee q \vee r) \wedge (\neg p \vee \neg r \vee s)$$

suas cláusulas seriam traduzidas para

- c_1 com lista: $\{p^1, q^1, r^1\}$
- c_2 com lista: $\{\overline{p}^2, q^2, r^2\}$
- c_3 com lista: $\{\overline{p}^3, \overline{r}^3, s^1\}$

Seja C o conjunto contendo todos os c_i criamos uma clique com $C \cup A$. Nosso grafo tem portanto a seguinte configuração(considere x' como \overline{x}):

Suponha a cláusula p, se p é verdadeiro então $a_p^{(1)}, a_p^{(2)}, a_p^{(3)}$ será colorido com p^1, p^2, p^3 , permitindo que a cor p^x possa, e que a cor p'^x não possa ser escolhidas para colorir uma cláusula.

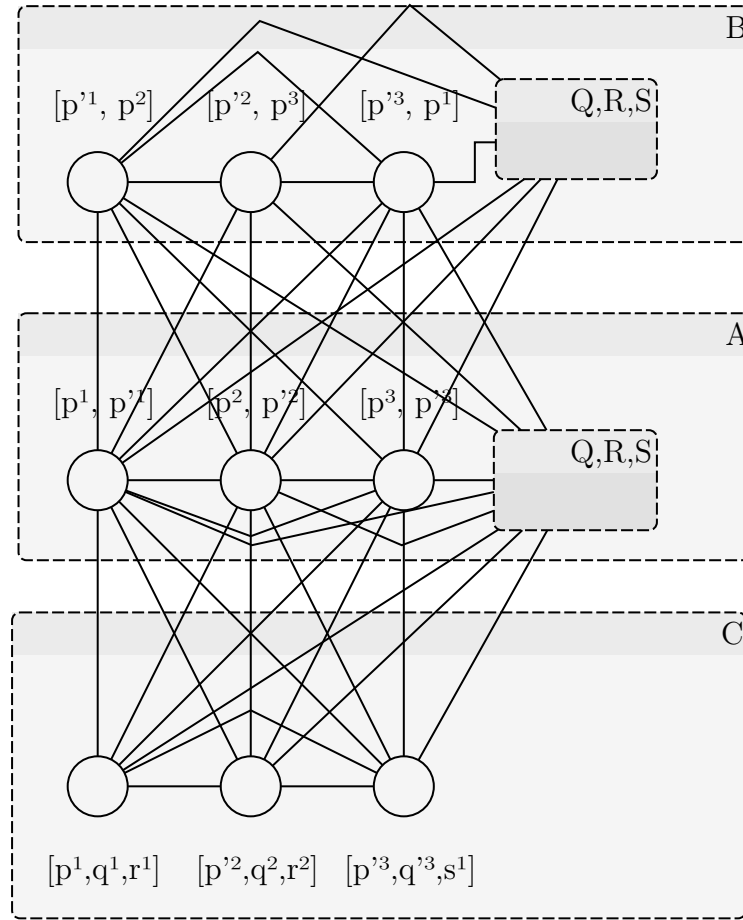


Figura 2.1: Grafo G: Transformação de 3-SAT em co-bipartido com foco na cláusula P

De tal forma, podemos facilmente notar que, expandindo a explicação anterior para os outros terminais uma resposta sim para o problema 3-SAT restrito nos leva a uma solução do problema de lista coloração em co-bipartido por exclusão das cores nas listas disponíveis. Em contrapartida a existência de uma lista coloração válida para o co-bipartido mostra uma solução para o 3-SAT restrito correspondente simplesmente descobrindo a representação em valor de literal das cores escolhidas para as cláusulas. \square

Corolário 3. *O problema de coloração é NP-Completo em $\text{Grafos}(0, 3)$.*

Demonstração. Tal resultado sai diretamente da prova dos Teoremas 4 e 5. \square

Portanto temos a dicotomia P/NP do problema de coloração em $\text{Grafos}(r, \ell)$.

$r \backslash \ell$	0	1	2	3	4	...	n
0	P	P	P	NP_c	NP_c	...	NP_c
1	P	P	NP_c	NP_c	NP_c	...	NP_c
2	P	NP_c	NP_c	NP_c	NP_c	...	NP_c
3	P	NP_c	NP_c	NP_c	NP_c	...	NP_c
4	NP_c	NP_c	NP_c	NP_c	NP_c	...	NP_c
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	NP_c
n	NP_c	NP_c	NP_c	NP_c	NP_c	...	NP_c

Tabela 2.3: Dicotomia P/NP_c do problema de coloração em Grafos(r, ℓ)

Capítulo 3

Análise parametrizada para coloração em Grafos(2,1)

Tendo mostrado a complexidade clássica de coloração em grafos(r, ℓ), é interessante elucidarmos quais características dos grafos(r, ℓ) se mostram propícias a abordagem parametrizada. A cardinalidade de suas partições, por exemplo, é um parâmetro interessante.

Nosso foco nesse capítulo será a classe dos grafos(2,1), já que a mesma é uma das classes com o menor número partes para o qual o problema já é NP-Completo.

Um Grafo(2,1) é um grafo para o qual seu conjunto de vértices po ser particionado em 2 conjuntos independentes e 1 clique, portanto ele nos entrega 3 parâmetros naturais, o tamanho da clique k , o tamanho do menor conjunto independente r_1 e o tamanho do maior conjunto independente r_2 .

3.1 O tamanho do menor conjunto independente como parâmetro

Em [5], Fellows (et. al) mostrou que o problema de lista coloração é $W[1]$ – *difícil* quando parametrizado pela cobertura de vértices, essa demonstração baseia-se em uma transformação do problema da clique multicolorida parametrizada pelo tamanho da clique buscada. Utilizaremos essa transformação para mostrar que coloração em grafos(2,1) é $W[1]$ -difícil..

Definição 8. *Clique multicolorida.*

Entrada: Um Grafo G com uma k -coloração própria.

Pergunta: Existe em G uma clique que contenha todas as k cores?

Teorema 6. *Coloração em Grafos(2,1) é $W[1]$ – difícil quando parametrizado pelo tamanho do menor conjunto independente.*

Demonstração. O problema da clique multicolorida é conhecidamente $W[1]$ – difícil [5]. A partir de uma instância G, k de clique multicolorida, iremos construir uma instância G' de lista coloração da seguinte forma:

- Para cada cor i presente em G cria-se em G' um vértice v_i (os chamaremos de vértices-cor).
- Para cada vértice u em G colorido com a cor i , adicionamos à lista do vértice-cor v_i em G' uma cor c_u relacionada a esse vértice (as chamaremos de cores-vértice).
- Para cada aresta $e(x, y) \notin E(G)$ onde $x, y \in V(G)$ cria-se em G' um vértice z_e adjacente ao vértice-cor v_i onde i representa as cores de x e y , a lista coloração de z_e será formada por c_x e c_y .

É notável que já que formamos um grafo bipartido, a cobertura por vértices é limitado por k . Perceba agora que se G possui uma clique multicolorida podemos facilmente colorir G' da seguinte forma:

Ao vértice-cor v_i atribua a cor-vértice c_u onde u é o vértice colorido com a cor i em G . Dessa forma todos os vértices z_e possuem ainda uma cor disponível para sua coloração já que ele representa uma não-aresta em G .

Agora observe que uma lista coloração válida em G' implica em uma clique multicolorida em G , isso se dá pois dois vértices x, y coloridos com cores diferentes em G não aparecem em uma lista de algum z_e em G' se e somente se existe uma aresta $e(x, y) \in E(G)$, portanto as cores-vértices escolhidas para os vértices v_i são uma respectivamente uma clique formadas por tais i em G . Mostramos assim que lista coloração parametrizada por cobertura por vértices é $W[1]$ – difícil.

Sabemos que coloração em Grafos(2,1) é equivalente a lista coloração em um grafo bipartido, portanto nossa tentativa de parametrizar a coloração de (2,1) pelo tamanho do menor conjunto independente é equivalente a parametrizar lista-coloração em bipartidos pelo tamanho da menor parte, mostrando assim que coloração em Grafos(2,1) parametrizada pelo tamanho do menor conjunto independente é $W[1]$ – difícil. \square

3.2 O tamanho do maior independente como parâmetro

Sabemos agora que parametrizar o problema pelo menor independente não é útil para produzirmos um algoritmo FPT, porém ao analisarmos o comportamento do problema quando parametrizado pelo maior independente vemos que a limitação do tamanho de r_2 também limita r_1 ; Tendo tal limitação a utilização de um método força bruta se mostra uma abordagem válida, como mostrado pelo teorema seguinte.

Teorema 7. *Coloração de Grafo(2,1) é FPT quando parametrizado pelo tamanho do maior conjunto independente.*

Demonstração. Sendo r_2 o tamanho do maior conjunto independente. Para colorir um grafo (2,1) são necessárias pelo menos k cores, onde k é a cardinalidade da parte clique. Para solucionar o problema, novamente usaremos a estratégia de transformar coloração de (2,1) em lista coloração de bipartido.

Em uma lista coloração de bipartido, se um vértice possui uma lista com mais cores do que o tamanho de sua vizinhança, ele sempre terá disponível uma cor para sua coloração, podemos portanto remover esse vértice do grafo sem alterar sua coloração, ao chegarmos ao ponto onde todo vértice com tal configuração foi removido, temos que tanto o número de vértices do grafo quanto o tamanho das listas de cada vértices estão limitados em função de r_2 , portanto rodar um algoritmo de força bruta para encontrar a coloração em tempo FPT. \square

3.3 O tamanho da clique como parâmetro

Para análise da complexidade parametrizada do problema, onde k , o tamanho da clique máxima, é o parâmetro voltaremos a observar coloração em um Grafo(2,1) como lista coloração em bipartido, dessa forma nosso problema parametrizado se torna lista coloração de bipartido parametrizado pelo número de cores da paleta.

Mostraremos que o tamanho da clique não é muito eficaz para a produção de um algoritmo FPT para o problema, na verdade, observamos que o problema permanece NP-difícil mesmo quando o tamanho da clique máxima é igual a três. Para tal se faz necessário definir o seguinte problema:

Definição 9. *PreColoring extension.*

Entrada: Um grafo G onde alguns vértices já possuem uma coloração definida com cores escolhidas dentre k possíveis cores.

Pergunta: É possível estender a k -coloração já existente para todo o grafo sem que dois vértices adjacentes possuam a mesma cor?

Teorema 8. ([9]) *PreColoring Extension em grafos bipartidos é NP-completo mesmo quando $k = 3$.*

Teorema 9. *lista 3-coloração em grafos bipartidos é NP-Completo.*

Demonstração. Suponha uma instância G, k do problema PreColoring Extension onde G é o grafo de entrada, e $k = 3$. Sabemos que existem vertices $v \in V(G)$ que já estão coloridos com uma cor $c_v \in 1, 2, 3$. Podemos ver tal configuração como um grafo G' onde os vértices v que estavam previamente coloridos possuem listas contendo apenas a c_v , e os demais vértices possuem lista igual a $\{1, 2, 3\}$, nos levando a um problema de lista 3-coloração.

Uma coloração possível para G implica em uma possível coloração para G' , já que nos basta atribuir aos vértices em G' as mesmas cores atribuídas em G . De forma análoga, uma lista coloração possível em G' implica em uma coloração possível em G . \square

Apesar do tamanho da paleta não ter se mostrado uma escolha adequada, ele levanta novos parâmetros que são interessantes para o problema de lista coloração em bipartidos, sabemos que lista coloração é polinomial se todos os vértices tem lista tamanho três (3-coloração), ou dois (2-coloração), e NP-Completo se tem listas de tamanho 1 à 3 [9], isso levanta duas formas de se abordar o problema

- O que acontece quando o número de vértices com listas de tamanho 1 e 2 é pequeno?
- O que acontece quando o número de vértices com listas de tamanho 3 é pequeno?

Mostraremos nas próximas seções como se dão tais comportamentos e como eles se relacionam a coloração de Grafos(r, ℓ).

3.4 A vizinhança da clique como parâmetro

Focaremos nesta seção em grafos(2,1) cuja a clique possui tamanho 3, já mostrada ser o menor tamanho onde o problema permanece difícil(como visto no Teorema 9 e em [9, 12]).

Observe que, se um nó não é vizinho de nenhum vértice na clique, após a redução esse vértice tem lista de tamanho três. Todo vértice vizinho a clique, após a transformação tem vértice com lista de tamanho $3 - x$ onde x é o número de seus vizinhos na clique. Um vértice nunca terá uma lista de tamanho 0 pois a clique é máxima.

Mostraremos que mesmo quando parametrizado pela quantidade de vértices com listas de tamanho um, dois, ou um e dois o problema é para-Np-completo. Para tanto é necessário encontrar uma instância do problema já parametrizado cuja solução permanece igualmente difícil.

Portanto essa seção será dividida em três casos, um contendo vértices de listas tamanho um, outro contendo vértices com listas de tamanhos dois, e finalmente contendo listas de tamanho um e dois.

3.4.1 Apenas vértices com listas de tamanho um ou três

É importante ressaltar que os seguintes teoremas estabelecem a base para a resolução do problema envolvendo os vizinhos da clique.

Teorema 10. *Lista 3-Coloração em grafos bipartidos é NP-completo mesmo quando todos vértices possuem listas de tamanho três, com exceção de três vértices com lista unitárias.*

Demonstração. Seja G uma instancia de Lista 3-Coloração em grafos bipartidos. Podemos transformar G em uma instância G' de Lista 3-Coloração em grafos bipartidos onde todos vértices possuem listas de tamanho três, com exceção de três vértices de listas unitárias. Além disso, G é uma instância sim se e somente se G' é uma instância sim.

A construção de G' é dada da seguinte forma:

- Faça $G' = G$;

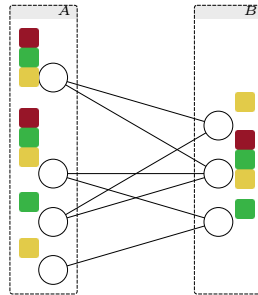
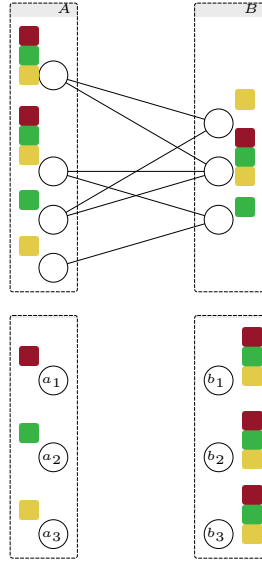
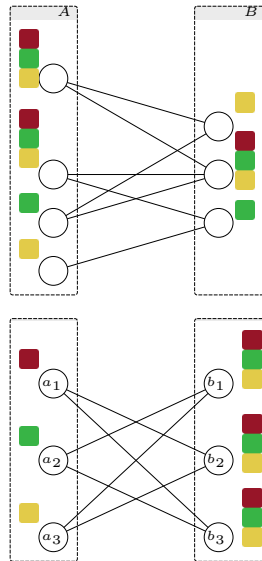


Figura 3.1: Grafo G' inicial.

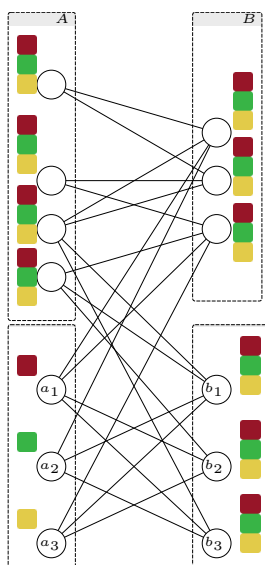
- adicione seis novos vértices em G' : a_1, a_2, a_3 e b_1, b_2, b_3 , onde as listas de $a_1, a_2, a_3, b_1, b_2, b_3$ são $\{1\}, \{2\}, \{3\}, \{1, 2, 3\}, \{1, 2, 3\}, \{1, 2, 3\}$;



- adicione em G' as arestas $(a_2, b_1), (a_3, b_1), (a_1, b_2), (a_3, b_2), (a_1, b_3), (a_2, b_3)$;



- Seja A, B a bipartição dos vértices de G .
 - para cada vértice $v \in A$ adicione uma aresta entre v e b_i , $i \in \{1, 2, 3\}$ se v não possui a cor i em sua lista;
 - para cada vértice $v \in B$ adicione uma aresta entre v e a_i , $i \in \{1, 2, 3\}$ se v não possui a cor i em sua lista;
 - para cada vértice em $A \cup B$ altere sua lista para $\{1, 2, 3\}$

Figura 3.2: Grafo G' final.

Observe que a_1, a_2, a_3 por possuírem listas unitárias devem ser coloridos com as cores 1, 2 e 3, respectivamente. Consequentemente, por construção, b_1, b_2 e b_3 terão que ser coloridos com as cores 1, 2 e 3, respectivamente (observe o gadget da Figura 3.3). Como os vértices a_1, a_2, a_3 e b_1, b_2, b_3 possuem cores pré-definidas, a adjacência dos demais vértices do grafo a estes vértices simula a proibição do uso de algumas cores em alguns vértices. Neste ponto, é fácil ver que G é uma instância sim se e somente se G' é uma instância sim. \square

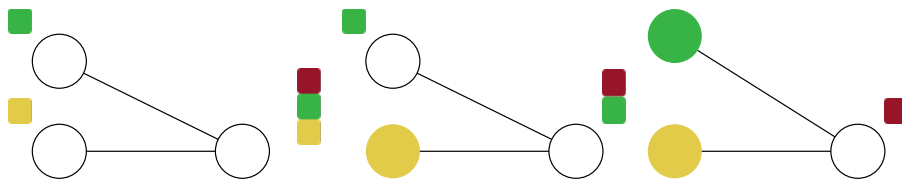


Figura 3.3: Gadget com vértices de lista um reproduzindo vértice de lista um em vértice de lista três

Interessantemente o problema é de trivial solução quando o número de vértices com listas tamanho um é zero, já que se torna o problema de 3-coloração em bipartidos, mas NP-Completo com 3 vértices, queremos portanto encontrar o menor número de vértices no qual o problema é NP-Completo, e consequentemente para-NP-Completo para nossa parametrização.

Lista 3-coloração em bipartido é trivial quando há apenas um vértice de lista unitária e todos os demais vértices possuem listas de tamanho três.

Teorema 11. *Lista 3-coloração em bipartido pode ser solucionado em tempo linear quando existem dois vértices com lista unitária e os demais vértices possuem listas de tamanho três.*

Demonstração. Para essa demonstração é necessária a observação em que existem duas possíveis configurações para essa instância:

- Ambos os vértices pertencem ao mesmo conjunto independente.
- Os vértices pertencem a conjuntos distintos.

No primeiro caso basta colorir os vértices de lista unitária com suas cores disponíveis e o conjunto independente ao qual pertencem com a cor de algum deles, colorindo o conjunto independente restante com a cor sobressalente.

No segundo caso, a coloração também é simples. Se tais vértices tem cores distintas basta colorir seus respectivos conjuntos com a mesma cor. Caso contrário, sem perda de generalidade podemos assumir que esses vértices não são adjacentes, senão a resposta do problema é negativa. Seja 1 a cor comum a esses vértices de lista unitária, neste caso, basta colorir o grafo obtido pela remoção desses vértices com as cores 2 e 3. \square

Dado os resultados apresentados nessa seção mostramos portanto que o número de vértices vizinhos a dois dos três vértices pertencentes a clique não é um parâmetro viável para uma solução FPT, já que a dificuldade permanece mesmo com 3 vértices.

3.4.2 Apenas vértices com listas de tamanho dois ou três

Mostraremos nessa seção que vértices com listas tamanho dois não são suficientes para que um algoritmo FPT seja extraído.

Como já visto o problema é de trivial solução quando todos os vértices tem listas de tamanho três, portanto precisamos ainda encontrar qual número de vértices de tamanho dois onde o problema se mantém NP-Completo.

Teorema 12. *Seis vértices com listas tamanho 2 são necessários e suficientes para que lista-coloração em bipartido seja NP-Completo.*

Demonstração. Para mostrarmos que qualquer número de vértices abaixo de 6 é insuficiente, mostraremos que a menos que existam pelo menos 3 vértices com distintas listas tamanho dois em cada conjunto independente, a coloração é simples de ser feita.

Se um conjunto independente contém apenas dois vértices com listas tamanho dois, podemos afirmar que todos os vértices nesse conjunto compartilham uma cor em suas listas, podendo colorir tal conjunto com essa cor, todos os outros vértices ainda têm pelo menos uma cor disponível para sua coloração podendo ser colorido com ela. É fácil notar que o argumento se estende para o caso onde alguma lista se repete dentro de qualquer r , já que dessa forma existe uma cor em comum entre todas as listas de tal r .

Para completar nossa demonstração basta portanto, encontrar uma configuração onde o problema de lista coloração permanece NP-Completo.

Para tanto nos é interessante agora a vizinhança entre os vértices com lista dois, iremos isolar as instâncias em alguns casos.

- Vizinhança de algum vértice é tamanho um:

Nesse caso podemos notar que independentemente do vértice $v \in r_1$ e seu vizinho $u \in r_2$, eles sempre compartilharão uma cor, colorimos o vértice u com tal cor, além disso como conhecemos a vizinhança de v sabemos que nenhum outro vértice é vizinho deste, podemos então colorir os restantes vértices de r_2 com a cor de v , dessa forma uma das três cores ainda resta e podemos a usar para colorir o restante do r_1 , independente das ligações entre os demais vértices de r_1 e r_2 .

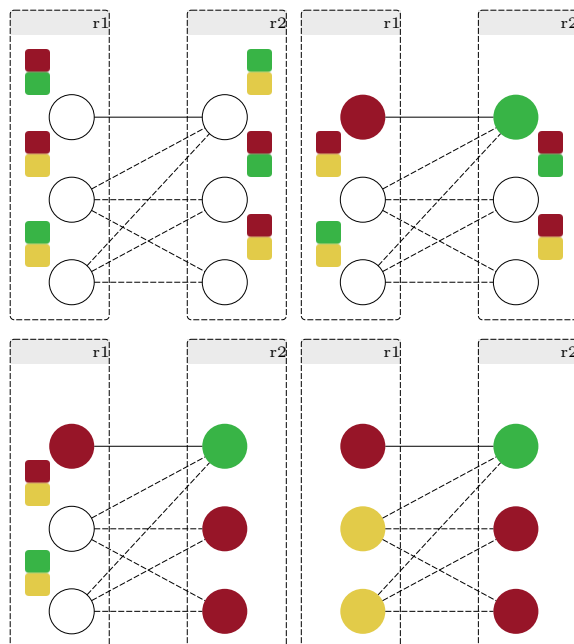


Figura 3.4: Demonstração de coloração para vizinhança de tamanho um.

- Um vértice têm vizinhança tamanho dois, e compartilha uma cor com ambos os vizinhos:

Aqui podemos executar o seguinte algoritmo, sem perda de generalidade pinte os vértices vizinhos à $v \in r_1$ com a cor compartilhada, novamente por conhecermos a vizinhança do vértice v podemos pintar ele e os demais vértices de r_2 com a cor restante, dessa forma ainda nos resta uma cor para colorir os demais vértices de r_1 .

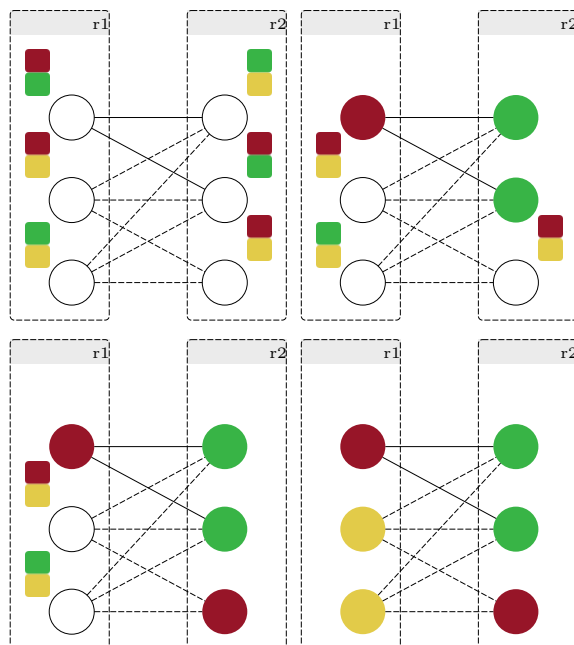


Figura 3.5: Demonstração de coloração para vizinhança de tamanho dois com cores compartilhadas.

- Todo vértice tem vizinhança de tamanho dois e nenhuma vizinhança possui uma cor em comum:

Observe que esse caso tem suas restrições derivadas dos casos acima, aqui mostraremos como PreColoring extension se reduz a esse caso, mostrando finalmente sua Np-Compleitude.

As restrições impostas a esse caso nos levam a uma única possível estrutura Γ onde suas duas possíveis colorações são intercambiáveis, vide figura 3.6. Portanto se mostra verdade que podemos excitar uma cor qualquer em outro vértice de lista tamanho três se o ligarmos a dois dos três vértices presentes no independente oposto sem ferir a bipartição do grafo.

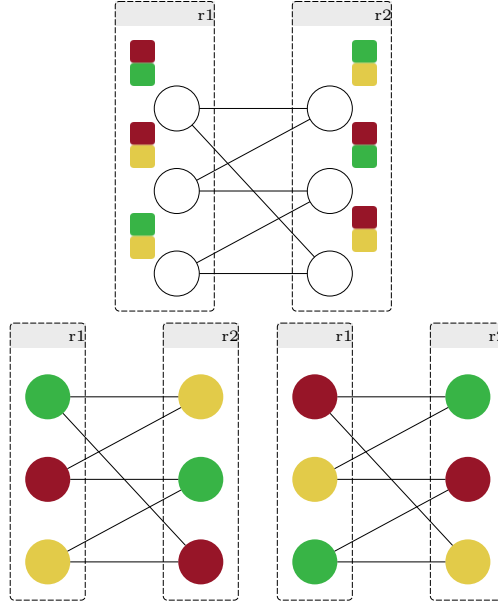


Figura 3.6: Estrutura Γ e suas possíveis colorações.

Assim sendo para reduzir um problema de PreColoring extension em bipartido Π ao nosso problema Π' basta que para todo vértice pré-colorido $v_{c_j} \in V(\Pi) | c_j \in C$ cria-se um paralelo $u \in V(\Pi')$ com lista de tamanho três e liga-o aos vértices de Γ a fim de excitar sua cor da seguinte forma:

Escolha sem perda de generalidade um vértice $v \in r1$ pré-colorido com a cor c_j , observe que Γ possui dois conjuntos de ligações capazes de excitar c_j em u utilizando o gadget da Figura 3.3, escolha um desses conjuntos. Portanto $\forall v_{c_j} | j \neq i$ basta realizar a ligação com Γ respeitando o conjunto de ligações já escolhido.

Os demais vértices de $V(\Pi)$ são construídos mantendo suas respectivas vizinhanças em Π' e com lista de tamanho três, é de pouca dificuldade compreender como uma resposta para Π' implica em uma resposta para Π , pois se Π' é lista colorível, então Π é colorível respeitando a pré-coloração, já que a coloração de Γ é indiferente para os vértices não vizinhos à ela e sempre respeita a coloração de sua vizinhança, para a demonstração da contraparte basta escolher as mesmas cores escolhidas em Π para seus respectivos em Π' que implicará em uma coloração para Γ inofensiva ao resultado.

□

É importante notar que para o acontecimento de haver vértices com listas tamanho um e dois, basta notar que podemos remover aqueles que contém listas de tamanho um, que propagará a remoção de sua cor às listas de vértices vizinhos, e ao realizar isso

iterativamente, acabaremos com um caso em que todos os vértices terão ou listas de tamanho dois ou três, caindo em algum dos casos supracitados.

3.5 Tamanho da não vizinhança da clique como parâmetro

Como visto na seção anterior, os vértices que não são vizinhos a clique, quando transformados em vértices do problema de lista coloração se transformam em vértices com listas tamanho três, portanto nosso desejo é resolver lista coloração em bipartidos com listas de tamanho um a três parametrizado pela quantidade de vértices com lista de tamanho três, a solução deriva do seguinte teorema.

Teorema 13. *Lista 3-coloração em bipartidos é FPT quando parametrizado pela quantidade de vértices com lista de tamanho três.*

Demonstração. Dado que temos k vértices com 3 escolhas cada é possível montar um algoritmo de busca em árvore de altura limitada de tamanho 3^k , e então executar o algoritmo linear proposto em [12] obtendo um algoritmo $\mathcal{O}(3^k n)$. \square

Capítulo 4

Conclusão

Apresentamos aqui o desenvolvimento e especificações da dificuldade do problema de coloração em grafos(r, ℓ). Ao nos aprofundar na investigação percebemos ainda a inesperada inaptidão da maior parte dos parâmetros relacionados ao tamanho das partições para a extração de um algoritmo FPT. Nas seções seguintes sumariaremos nossos resultados e mostraremos consequências e questionamentos relacionados.

4.1 Resultados e consequências

Obtivemos no capítulo 2 uma dicotomia P/NP_c para o problema de coloração em grafos(r, ℓ), é conhecido que o problema de coloração em um grafo G pode ser visto como um problema de clique cover em seu complemento G' [6]. DAssim sendo podemos estender a dicotomia P/NP_c da tabela 2.3 para o problema de clique cover em grafos(r, ℓ) simplesmente trocando as linhas pelas colunas da tabela, dessa forma obtemos a seguinte dicotomia P/NP_c :

$r \backslash l$	0	1	2	3	4	...	n
0	P	P	P	P	NP_c	...	NP_c
1	P	P	NP_c	NP_c	NP_c	...	NP_c
2	P	NP_c	NP_c	NP_c	NP_c	...	NP_c
3	NP_c	NP_c	NP_c	NP_c	NP_c	...	NP_c
4	NP_c	NP_c	NP_c	NP_c	NP_c	...	NP_c
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	NP_c
n	NP_c	NP_c	NP_c	NP_c	NP_c	...	NP_c

Tabela 4.1: Dicotomia P/NP_c do problema de clique cover em Grafos(r, ℓ)

Já no capítulo 3 exploramos com profundidade o comportamento do problema em $\text{Grafos}(2, 1)$. Obtendo dessa forma 2 algoritmos FPT, uma demonstração de $W[1]$ – *difficuldade* e algumas de para-NP-completude, tendo colateralmente mostrado resultados clássicos e parametrizados para o problema de lista coloração em bipartidos.

4.2 Trabalhos futuros

Alguns questionamentos levantados durante a produção deste trabalho que tomaram forma de possíveis trabalhos futuros foram:

- Existe uma relação entre as parametrizações da coloração de $\text{Grafos}(2, 1)$ e clique cover de $\text{Grafos}(1, 2)$?
- Quais são as características que afetam parametrizações em $\text{Grafos}(r, \ell)$ diferentes de $\text{Grafos}(2, 1)$?
- Existe algum parâmetro que seja possível extrair um algoritmo FPT para qualquer r ou ℓ ?

Referências

- [1] APPEL, K., HAKEN, W. The Solution of the Four-Color-Map Problem. *Scientific American* 237, 4 (1977), 108–121.
- [2] BOLLOBÁS, B. *Modern Graph Theory*. 1998, p. 165–166.
- [3] CHARTRAND, G., ZHANG, P. *A First Course in Graph Theory*. 2012, p. 189–190.
- [4] DOWNEY, R. G., FELLOWS, M. R., REGAN, K. W. Parameterized circuit complexity and the W hierarchy.
- [5] FELLOWS, M., FOMIN, F. V., LOKSHTANOV, D., ROSAMOND, F., SAURABH, S., SZEIDER, S., THOMASSEN, C. On the Complexity of Some Colorful Problems Parameterized by Treewidth.
- [6] GAREY, M. R., JOHNSON, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. 1979.
- [7] JANSEN, K. Complexity Results for the Optimum Cost Chromatic Partition Problem.
- [8] JANSEN, K., SCHEFFLER, P. Generalized coloring for tree-like graphs.
- [9] KRATOCHVIL, J. PreColoring Extension With Fixed Color Bound.
- [10] KÖNIG, D. "Gráfok és mátrixok". *Matematikai és Fizikai Lapok*. 1931, p. 116–119.
- [11] LEWIS, R. *A Guide to Graph Colouring: Algorithms and Applications*. 2015.
- [12] M.HUJTER, ZS.TUZA. PreColoring Extension. II. Graph classes related to bipartite graphs.