UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE MATEMÁTICA

PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA

MESTRADO ACADÊMICO EM MATEMÁTICA

LÚCIO CARLOS PIMENTEL PAIVA

COMPLEXIDADE DO PROBLEMA DE K-COLORAÇÃO EM GRAFOS LIVRES DE H

FORTALEZA

2019

LÚCIO CARLOS PIMENTEL PAIVA

COMPLEXIDADE DO PROBLEMA DE K-COLORAÇÃO EM GRAFOS LIVRES DE H

Dissertação apresentada ao Curso de Mestrado Acadêmico em Matemática do Programa de Pós-Graduação em Matemática do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Matemática. Área de Concentração: Combinatória

Orientadora: Prof. Dra. Ana Shirley Ferreira da Silva

FORTALEZA

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

P169c Paiva, Lúcio Carlos Pimentel.
Complexidade do Problema de K-Coloração Em Grafos Livres de H / Lúcio Carlos Pimentel Paiva. – 2023.
53 f.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Matemática, Fortaleza, 2023.
Orientação: Profa. Dra. Ana Shirley Ferreira da Silva.

1. Coloração. 2. Índice cromático. 3. Grafos livres de P_k. 4. Complexidade. I. Título.
CDD 510

LÚCIO CARLOS PIMENTEL PAIVA

COMPLEXIDADE DO PROBLEMA DE K-COLORAÇÃO EM GRAFOS LIVRES DE H

Dissertação apresentada ao Curso de Mestrado Acadêmico em Matemática do Programa de Pós-Graduação em Matemática do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Matemática. Área de Concentração: Combinatória

Aprovada em: 30/08/2019

BANCA EXAMINADORA

Prof. Dra. Ana Shirley Ferreira da Silva
(Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Júlio César Silva Araújo
Universidade Federal do Ceará (UFC)

Prof. Dr. Carlos Vinícius Gomes Costa Lima
Universidade Federal do Rio de Janeiro (UFRJ)

Dedico este trabalho aos meus familiares e amigos.

AGRADECIMENTOS

À minha família, em especialmente a minha amada mãe Rita Lúcia Pimentel e os meus irmãos Ana Carla Pimentel Paiva e Carlos Henrique Pimentel Paiva, por todo amor, carinho, conselhos e incentivo aos meus estudos. Por acompanhar meu desenvolvimento profissional e acadêmico. E, por acreditar, incondicionalmente, no meu potencial.

À minha orientadora pela ajuda em todas as etapas deste trabalho.

Aos membros da banca Prof. Dr. Júlio César Silva Araújo e Prof. Dr. Carlos Vinícius Gomes Costa Lima pela disponibilidade.

A todos os professores que foram cruciais em minha formação e aos meus amigos pelos inúmeros momentos e ensinamentos compartilhados.

À Andrea Costa Dantas, secretária da pós-graduação, pela presteza e competência.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

“Alguns homens vêem as coisas como são, e dizem: Por que? Eu sonho com coisas que nunca foram e digo: Por que não?"

(SHAW, 1922, p. 7)

RESUMO

O problema de coloração é um dos problemas mais estudados em Teoria dos Grafos. É sabido que decidir se um grafo possui uma coloração com 3 cores é um problema NP-completo, mesmo quando se impõem restrições sobre o grafo, como por exemplo, grafos planares ou grafos linha. Um teorema conhecido, de autoria de Král et al., acerca da dificuldade do problema diz que, dados um grafo G livre de H e um inteiro k, decidir se o número cromático de G é no máximo k é polinomial quando H é um subgrafo induzido de P_4 ou de $P_3 + K_1$, e é NP-completo, caso contrário. Isso motivou o estudo da dificuldade do problema para valores fixos de k. Em particular, quando H é um grafo conexo, os únicos casos ainda em aberto ocorrem quando H é um caminho.

Neste trabalho, apresentamos uma revisão bibliográfica do problema. Além disso, mostramos dois dos principais resultados relacionados a grafos livres de caminhos.

Palavras-chave: coloração; índice cromático; grafos livres de P_k; complexidade.

ABSTRACT

The coloring problem is one of the most studied problems in Graph Theory. It is known that deciding whether a graph has a coloring with 3 colors is an NP-complete problem, even when constraints are imposed on the graph, such as flat graphs or line graphs.
A well-known theorem, authored by Král et al., about the difficulty of the problem says that, given a free graph G of H and an integer k, deciding whether the chromatic number of G is at most k is polynomial when H is an induced subgraph of P_4 or $P_3 + K_1$, and is NP-complete otherwise. This motivated the study of the difficulty of the problem for fixed values of k. In particular, when H is a connected graph, the only remaining open cases occur when H is a path. In this work, we present a bibliographical review of the problem. Furthermore, we show two of the main results related to path-free graphs.

Keywords: coloring; chromatic index; free graphs of P_k; complexity.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

SUMÁRIO

1 INTRODUÇÃO

Em 1852, Francis Guthrie conjecturou que quatro cores seriam suficientes para colorir qualquer mapa desenhado sobre um plano de forma que regiões adjacentes tivessem cores diferentes (MACKENZIE, 2004). Trata-se do primeiro problema conhecido de coloração de grafos. Guthrie não conseguiu provar sua conjectura, sobre a qual vários matemáticos se debruçaram na época, como De Morgan e Cayley (CAYLEY, 1879). Kempe apresentou uma primeira tentativa de demonstração para a conjectura em 1879, que foi tida como válida até 1890, ano em que Heawood conseguiu apontar um erro na prova (HEAWOOD, 1890). Apesar do erro, Kempe introduziu o conceito de cadeias de Kempe, que pôde ser usado por Heawood para demonstrar o Teorema das cinco cores. Kempe também concebeu o método dos descarregamentos. Este acabou sendo o método utilizado na primeira prova definitiva da conjectura, obtida por Appel e Haken apenas em 1976 (APPEL K.; HAKEN, 1977a)(APPEL K.; HAKEN, 1977b), tendo sido também a primeira prova de um teorema em matemática assistida por computador. Desde então, a conjectura passou a ser chamada de Teorema das quatro cores. Apesar dos avanços obtidos por Robertson, Sanders, Seymour e Thomas (ROBERTSON N.; SANDERS, 1977), não há hoje prova conhecida do Teorema das quatro cores que possa ser verificada manualmente.

O problema da coloração não se restringe aos grafos planares e hoje é estudado sob diversas variações, como a coloração de arestas, a coloração por listas e a extensão de pré-colorações. Cabe observar aqui que, além de se relacionar com outras áreas da matemática, trata-se de um problema com diversas aplicações práticas, como alocação de frequências em telecomunicações (AARDAL K.; HOESEL, 2007) e alocação de registros de computadores (CHAITIN G. ; AUSLANDER, 1981). Apesar de se tratar de um problema com muitas aplicações, determinar se um grafo G pode ser colorido com k cores é um problema inerentemente custoso do ponto de vista computacional.

Em 1971, Stephen Cook demonstrou a existência de um problema NP-completo e formalizou o conceito de redução polinomial entre problemas de decisão (COOK, 1971). No ano seguinte, Richard Karp demonstrou em um famoso trabalho a NP-completude de 21 problemas, dos quais a coloração faz parte (KARP, 1972). O estudo do problema da coloração do ponto de vista de complexidade computacional ganhou grande interesse já nos anos que se seguiram, com publicações como (STOCKMEYER, 1973), na qual Larry Stockmeyer demonstrou a NP-completude da 3-coloração para grafos planares, e (JOHNSON D.; GAREY; STOCKMEYER,

1976), na qual Michal Garey, David Johnson e Larry Stockmeyer forneceram, entre outros resultados, uma prova simplificada da NP-completude da 3-coloração para grafos quaisquer.

A dificuldade em atacar o problema da coloração é notória. Mesmo para grafos planares (que possuem número cromático limitado a 4) ou grafos cujos vértices possuem grau no máximo igual a 4 (GAREY M. R. ; JOHNSON, 1979), o problema da 3-coloração é NP-completo. A k-coloração de grafos linha também é um problema NP-completo, pois a determinação do índice cromático de um grafo também o é (HOLYER, 1981). Mesmo a aproximação do número cromático possui suas barreiras: é NP-difícil aproximar o número cromático de grafos com um fator $n^{(1-\varepsilon)}$, para todo $\varepsilon > 0$ (ZUCKERMAN, D., 2006).

Sendo o problema de coloração NP-completo e difícil de tratar de maneira aproximada, uma possível abordagem é analisar o problema restrito a alguma classe de grafos. Esse tipo de estudo é comum em problemas NP-completos, dado que em alguns casos a restrição a uma classe de instâncias pode ser resolvida em tempo polinomial. A título de exemplo, o problema da coloração de grafos perfeitos admite solução polinomial (GROTSCHEL, M.; LOVASZ, L.; SCHRIJVER, A., 1984). Um caso interessante ocorre quando se toma um grafo H e queremos classificar o problema de coloração restrito à classe dos grafos livres de H (ou seja, que não possuem H como subgrafo induzido), conforme o teorema abaixo.

Teorema 1.0.1 *(KRÁL; KRATOCHVÍL; WOEGINGER, G., 2001) Dados um grafo G, livre de H previamente fixado, e um número natural k, decidir se $\chi(G) \leq k$ é um problema polinomial quando H é um subgrafo induzido de P_4 ou de $P_3 + K_1$, e é NP-completo caso contrário.*

Podemos, em um próximo passo, fixar o valor de k em vez de deixá-lo fazer parte da entrada do problema de decisão. A ideia é controlar a complexidade do algoritmo através desse parâmetro. Se a complexidade para resolver um problema de decisão qualquer depende de dois parâmetros n e k e conseguimos encontrar, por exemplo, um algoritmo com complexidade $O(n^k)$, isso significa que quando k é um valor fixo, por exemplo 3, resolvê-lo passa a ser considerado polinomial, pois a complexidade passa a ser $O(n^3)$. Observe que, em geral, isso não é possível devido aos resultados mencionados. Desta forma, é necessário também fazer restrições acerca do tipo de grafo G. Alguns resultados importantes nesse sentido são citados em (HUANG, S., 2016), como o teorema que segue.

Teorema 1.0.2 *(VADIM V. L.; KAMINSKI M., 2007) Para todo $k \geq 3$, o problema da k-coloração é NP-completo sobre o conjunto dos grafos com cintura maior do que ou igual a g, para $g \geq 3$.*

Como consequência imediata, o problema da k-coloração é NP-completo sobre a classe dos grafos livres de H, sempre que H contiver um ciclo. Mais do que isso, Ian Holyer deduz em (HOLYER, 1981) que o problema da k-coloração de arestas é NP-completo para todo $k \geq 3$. Como a coloração das arestas de um grafo G é equivalente à coloração do grafo linha de G, denotado $L(G)$, é imediato que o problema da coloração sobre o conjunto dos grafos linha é NP-completo. Este resultado tem como consequência a NP-completude do problema da k-coloração de grafos livres de $K_{1,3}$ para todo $k \geq 3$, pois todos os grafos linha são livres de $K_{1,3}$. Logo, os possíveis casos em que a k-coloração de um grafo G livre de H pode ser determinada em tempo polinomial ocorrem quando H é uma floresta linear, pois H não pode conter ciclos ou vértices com grau maior do que dois. Em particular, se impusermos que H seja conexo, temos $H = P_t$ para algum t. Os resultados existentes até o momento da escrita são resumidos na Tabela 1.

Tabela 1 – Complexidade da k-coloração de grafos livres de P_t

	$k = 3$	$k = 4$	$k = 5$	$k \geq 6$
$t \leq 5$	P [1]	P [1]	P [1]	P [1]
$t = 6$	P [2]	?	NP-c [3]	NP-c [3]
$t = 7$	P [4], [5]	NP-c [3]	NP-c	NP-c [6]
$t \geq 8$	?	NP-c	NP-c	NP-c

[1] (HOÀNG C.T.; KAMIŃSKI, 2010)

[2] (RANDERATH B. ; SCHIERMEYER, 2004)

[3] (HUANG, S., 2016)

[4] (CHUDNOVSKY M.; MACELI, 2014)

[5] (CHUDNOVSKY M.; MACELI, 2015)

[6] (BROERSMA H.; FOMIN, 2013)

Os casos que ainda estão em aberto são $k = 3$ e $t \geq 8$; e $k = 4$ e $t = 6$. Note que todo grafo livre de P_t também é livre de $P_{t'}$, para todo $t' > t$. Logo, um resultado polinomial

para uma combinação (k,t) pode ser extendida a valores $t' < t$; por outro lado, resultados de NP-completude podem ser extendidos a valores $t' > t$.

No Capítulo 2, definimos os conceitos necessários ao entendimento do texto; no Capítulo 3 fazemos uma revisão do estado da arte e comentamos as perspectivas da pesquisa sobre o tema; no Capítulo 4, exploramos um caso em que o problema é FPT; no Capítulo 5, reproduzimos a prova de um dos casos de NP-completude; no Capítulo 6, fazemos algumas considerações finais.

2 PRELIMINARES

Neste capítulo, introduzimos as definições básicas de teoria dos grafos, de complexidade computacional e de complexidade parametrizada.

2.1 Grafos

Iniciaremos com a definição de métrica riemanniana e a partir dessa definiremos variedades imersas e imersões isométricas.

Definição 2.1.1 *(Grafo) Um grafo é um par (V,E), onde V é um conjunto não vazio, chamado conjunto de vértices, e E é um multiconjunto de pares ordenados em $V \times V$, chamado conjunto de arestas. Para um grafo G dado, podemos escrever $V(G)$ para o conjunto de vértices e $E(G)$ para o conjunto de arestas. Usaremos as notações $e = (u,v) \in E$ ou $e = uv$ conforme a conveniência.*

Um *laço* é uma aresta $e = uu$, e uma aresta é *múltipla* se aparece mais de uma vez em E. Um grafo para o qual $uv \neq vu$ é dito um *digrafo* ou *grafo direcionado*; caso a ordem dos pares que definem as arestas não seja importante, o grafo é dito *não direcionado*. Um grafo não direcionado sem laços ou arestas múltiplas é dito *simples*. Um grafo $G = (V,\emptyset)$ é dito *vazio*. No restante do texto, fazemos um abuso de linguagem e nos referimos a grafos simples apenas como grafos.

Definição 2.1.2 *(Vizinhos/Vizinhança) Seja G um grafo. Se $e = uv \in E(G)$, dizemos que u e v são adjacentes ou vizinhos. A aresta e é incidente a u e v, e esses vértices são chamados extremidades de e. A vizinhança $N(u)$ de um vértice u é o conjunto de todos os vizinhos de u. A vizinhança fechada de u é dada por $N[u] = N(u) \cup \{u\}$. O grau de um vértice u, denotado por $d(u)$, é a quantidade de arestas incidentes em u. Duas arestas $e_1, e_2 \in E(G)$ são ditas adjacentes se compartilham um vértice.*

Note que se G é um grafo simples, vale $d(u) = |N(u)|$. O *grau máximo* e o *grau mínimo* de G são dados respectivamente por $\Delta(G) = \max_{v \in V(G)} d(v)$ e $\delta(G) = \min_{v \in V(G)} d(v)$. O *grau médio* de um grafo G é dado por $\overline{d}(G) = \frac{1}{|V(G)|} \sum_{v \in V(G)} d(v) = \frac{2|E(G)|}{|V(G)|}$.

Definição 2.1.3 *(Subgrafo) Um grafo H é chamado subgrafo de G se $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. Dado $X \subseteq V(G)$, o subgrafo induzido por X é $G[X] = (X,E')$ tal que, para todo $u,v \in X$,*

tem-se $uv \in E'$ se e somente se $uv \in E(G)$. Definimos também

$$N(X) = \left(\bigcup_{v \in X} N(v) \right) \setminus X$$

e $N_X(v) = N(v) \cap X$. Um conjunto de vértices $X \subseteq V(G)$ é dito dominante se $V(G) = X \cup N(X)$, ou seja, todo vértice de G está em X ou possui um vizinho em X.

Definição 2.1.4 *(Complemento de um grafo) Seja G um grafo simples. O complemento de G, denotado $\overline{G}$, é um grafo com $V(\overline{G}) = V(G)$ e $uv \in E(\overline{G})$ se e somente se $uv \notin E(G)$ $\forall u, v \in V(G)$.*

Um grafo simples G é *completo* se $uv \in E(G)$ $\forall u, v \in V(G)$. O grafo completo com k vértices é denotado por K_k. Um subconjunto $X \subseteq V(G)$ é uma *clique* se $G[X]$ é completo, e é um *conjunto independente* ou *conjunto estável* se $G[X] = (X, \emptyset)$. O tamanho (em número de vértices) da maior clique de G se escreve como $\omega(G)$, e o tamanho do maior conjunto independente de G se escreve como $\alpha(G)$. Se os vértices de G podem ser divididos em dois conjuntos independentes, dizemos que G é *bipartido*. Se G é bipartido e cada vértice de um dos conjuntos independentes é vizinho de todos os outros vértices do outro conjunto independente, dizemos que G é *bipartido completo*. O grafo bipartido completo com partes de tamanho p e q é denotado por $K_{p,q}$.

2.2 Caminhos

Definição 2.2.1 *Um caminho em um grafo G é uma sequência de vértices $(v_1, v_2, \ldots, v_k)$ tal que $v_i v_{i+1} \in E(G)$ $\forall i \in \{1, 2, \ldots, k-1\}$. Os vértices v_1 e v_k são as extremidades do caminho. Um ciclo em G é definido de maneira similar; a diferença é que nesse caso se impõe $v_k = v_1$. O caminho com k vértices é denotado por P_k enquanto que o ciclo com k vértices é denotado por C_k.*

Chama-se a *cintura* de G o comprimento do menor ciclo contido em G. Caso nenhum subgrafo de G seja um ciclo, diz-se que a cintura de G é infinita.

2.3 Conexidade

Definição 2.3.1 *(Grafo Conexo) Um grafo G é conexo se para todo par de vértices $u, v \in V(G)$ existe um caminho de extremidades u e v. Ademais, G é 2-conexo se $G - v$ é conexo para todo $v \in V(G)$.*

Um grafo que não contém ciclos é chamado de *floresta*; um grafo conexo que não contém ciclos é chamado de *árvore*. Uma árvore na qual no máximo um vértice tem grau maior do que 1 é chamada de *estrela*. Uma floresta H é dita *linear* se $\Delta(H) \leq 2$, ou seja, se H é uma união disjunta de caminhos.

Definição 2.3.2 *Dados dois grafos G e H, definimos a soma disjunta de G e H, denotada por $G+H$, como o grafo com $V(G+H) = V(G) \cup V(H)$ e arestas $E(G+H) = E(G) \cup E(H)$. Ou seja, o grafo $G+H$ é constituído por uma cópia de G e uma cópia de H.*

Definição 2.3.3 *(Isomorfismo de Grafos) Dois grafos G_1 e G_2 são isomorfos se existe uma bijeção $f : V(G_1) \to V(G_2)$ tal que $v_1v_2 \in E(G_1)$ se e somente se $f(v_1)f(v_2) \in E(G_2)$. Um grafo G é dito livre de um grafo H se não possui subgrafo induzido isomorfo a H. Um grafo G é dito livre de uma família de grafos $\mathscr{H}$ se G é livre de H para todo grafo $H \in \mathscr{H}$.*

2.4 As classes P e NP

As definições relativas à presente seção podem ser encontradas em (GAREY M. R. ; JOHNSON, 1979).

Definição 2.4.1 *(Problema de Decisão)*

Um problema de decisão Π é dado por um conjunto de instâncias D_Π e um conjunto $Y_\Pi \subseteq D_\Pi$ de instâncias positivas.

No caso do problema de k-coloração, as instâncias são grafos e as instâncias positivas são os grafos que admitem uma k-coloração.

Definição 2.4.2 *(Linguagem) Uma linguagem L sobre um conjunto finito Σ, denominado alfabeto, (cujos elementos são chamados símbolos) é um subconjunto de Σ^*, conjunto formado por sequências finitas de símbolos de Σ mais um símbolo especial ε, que corresponde à sequência vazia.*

Uma codificação e é uma representação das instâncias $I \in \Pi$ dentro da linguagem L. Em (GAREY M. R. ; JOHNSON, 1979), são feitas algumas considerações sobre as codificações "razoáveis". Para todos os efeitos, uma codificação razoável transforma cada instância $I \in D_\Pi$ do problema original em uma sequência de símbolos $I^* \in \Sigma^*$ cujo *comprimento* $|I^*|$ (número de símbolos) é limitado por uma função polinomial dos parâmetros de interesse (no caso do objeto

de estudo presente, o número de vértices e arestas de um grafo G). Chamaremos o conjunto das codificações dos elementos de Y_Π por e de $L[\Pi, e]$.

Definição 2.4.3 *(Máquina de Turing) Uma máquina de Turing determinística consiste de: um conjunto de símbolos* Γ*, incluindo um subconjunto* $\Sigma \subset \Gamma$ *de símbolos de entrada e um símbolo "em branco"* $b \in \Gamma - \Sigma$*; uma fita de memória infinita onde são escritos os símbolos, inicialmente contendo b em todas as posições; um conjunto de estados Q, incluindo um estado inicial* q_0 *e dois estados de parada* q_Y *e* q_N*; e uma função de transição* $\delta : (Q - \{q_Y, q_N\}) \times \Gamma \to Q \times \Gamma \times \{-1, 1\}$.

A operação de uma máquina de Turing determinística sobre uma linguagem L é simples: os símbolos correspondentes à codificação da instância do problema são escritos sobre a fita, a partir de uma posição inicial fixa (posição zero da fita de memória). A máquina opera passo-a-passo, conforme o estado atual. Se o estado for $q = q_Y$ ou $q = q_N$, o programa pára, dando uma resposta "sim" ou "não" para a instância, respectivamente. Caso contrário, a máquina lê o símbolo $s \in \Gamma$ na posição atual da fita, e o valor de $\delta(q, s) = (q', s', \Delta)$ é calculado. O símbolo s é substituído na posição atual da fita por s', e a máquina se move uma posição para a esquerda se $\Delta = -1$ ou para a direita se $\Delta = 1$. O estado atual da máquina então muda para q'. Uma máquina *resolve* o problema Π se atinge o estado q_Y para todo elemento de $L[\Pi, e]$ e atinge q_N caso contrário. O *tempo de solução* $T_M(I^*)$ de uma instância $I^* \in L[\Pi, e]$ é o número de transições executadas pela máquina de Turing para parar.

Um problema (ou mais especificamente a linguagem L) *pertence à classe P* se existem uma máquina de Turing determinística M e um polinômio p tal que $\forall I^* \in \Pi,\ T_M(I^*) \leq p(|I^*|)$. Diz-se que M *resolve* Π em *tempo polinomial*. Se a codificação for "razoável", a codificação do problema não é relevante para efeitos de classificação de complexidade[1], e podemos atacar o problema original em vez de procurar uma representação de um algoritmo em uma máquina de Turing (GAREY M. R. ; JOHNSON, 1979)

Definição 2.4.4 *(Testemunha) Uma testemunha é um objeto associado a uma instância positiva* $I \in Y_\Pi$ *de um problema* Π *que permite verificar que de fato* $I \in Y_\Pi$.

Um problema *pertence a NP* se existe uma máquina de Turing determinística M tal que, para cada $I \in Y_\Pi$, existe uma testemunha T e M consegue verificar a partir de I e T que $I \in Y_\Pi$ em tempo polinomial. No caso do problema de k-coloração, definido na subseção seguinte,

[1] caso contrário seria necessário especificar que $L[\Pi, e] \in P$ em vez de apenas $\Pi \in P$

uma testemunha para um grafo G k-colorível poderia ser uma k-coloração própria de G. Uma definição usando uma *máquina de Turing não-determinística* está disponível em (GAREY M. R. ; JOHNSON, 1979).

Definição 2.4.5 *(Redução Polinomial) Uma redução polinomial de uma linguagem $L_1 \subseteq \Sigma_1^*$ para uma linguagem $L_2 \subseteq \Sigma_2^*$ é uma função $f : \Sigma_1^* \to \Sigma_2^*$ tal que existe uma máquina de Turing determinística que calcula f em tempo polinomial e para todo $x \in \Sigma_1^*$, $x \in L[\Pi_1, e_1]$ se e somente se $f(x) \in L[\Pi_2, e_2]$.*

Um problema Π é *NP-difícil* se existe uma redução polinomial de Π' para Π para todo $\Pi' \in NP$; além disso, ele é *NP-completo* se pertence a NP e é NP-difícil. A existência de um problema NP-completo foi provada em 1971 (COOK, 1971). A partir do primeiro problema NP-completo provou-se que outros problemas são NP-completos, uma vez que eles formam uma classe de equivalência de problemas de decisão (GAREY M. R. ; JOHNSON, 1979). Na seção 3.2 prova-se que a 3-coloração é um problema NP-completo a partir do 3SAT. Podemos estender o conceito de problema NP-difícil para outros tipos de problemas (otimização, por exemplo).

3 O PROBLEMA DE COLORAÇÃO

Neste capítulo exploraremos desde a básica definição de coloração até uma visão abrangente sobre a complexidade de coloração de grafos.

3.1 Coloração: Conceitos Básicos

Definição 3.1.1 *(Coloração de um Grafo) Uma k-coloração própria ou simplesmente k-coloração de um grafo G é uma função $c : V(G) \to S$, com $|S| = k$, tal que $c(v_i) \neq c(v_j)$ para toda aresta $v_iv_j \in E(G)$. Dizemos que G é k-colorível quando tal coloração existe.*

Salvo menção contrária, supõe-se que $S = \{1, \ldots, k\}$. Neste caso, usamos a notação $[k]$ para denotar o conjunto $\{1, \ldots, k\}$. Note que um grafo G que possui um laço não admite coloração, e que arestas múltiplas não restringem mais as colorações possíveis do que arestas simples. É por isso que, ao investigar o problema de coloração, faz-se a restrição a grafos simples.

Definição 3.1.2 *Define-se o número cromático de G como o menor valor de k para o qual G admite uma k-coloração própria; denotamos por $\chi(G)$. Se $\chi(G) \leq k$, então G é k-colorível. Se $\chi(G) = k$, dizemos que G é k-cromático; se além disso $\chi(G - v) < k$ para todo $v \in V(G)$, dizemos que G é k-crítico.*

Dados um conjunto S e um natural k, denotamos por $\binom{S}{k}$ os subconjuntos de S com k elementos.

Sejam G um grafo, $W = \{w_1, \ldots, w_r\} \subseteq V(G)$ e S um conjunto. Definimos uma *atribuição de listas de tamanho k sobre W*, ou simplesmente k-atribuição sobre W, como uma função $\mathscr{L} : W \to \binom{S}{k}$ que associa a cada vértice $w \in W$ uma *paleta* ou *lista* $\mathscr{L}(w)$. Dada uma k-atribuição $\mathscr{L}$ sobre $V(G)$, definimos como uma *k-coloração de G por listas*, ou ainda uma *$\mathscr{L}$-coloração* sobre G, uma k-coloração c tal que $c(v_i) \in \mathscr{L}(v_i)$ para todo $v_i \in V(G)$.

Dizemos que um grafo G é *k-escolhível* quando admite uma $\mathscr{L}$-coloração para toda k-atribuição $\mathscr{L}$ sobre $V(G)$. O *número de escolha* (ou *número cromático por listas*) de um grafo G é o menor inteiro k para o qual G é k-escolhível; é denotado por $ch(G)$. Como uma coloração por listas é também uma coloração própria, tem-se $ch(G) \geq \chi(G)$.

Uma *k-coloração das arestas de G* é uma função $f : E(G) \to S$, com $|S| = k$, tal que $f(e_i) \neq f(e_j)$ se as arestas e_i e e_j são adjacentes. O menor número para o qual um grafo G

admite k-coloração das arestas é chamado *índice cromático* e denotado $\chi'(G)$. Pelo Teorema de Vizing (VIZING, 1964), sabe-se que $\Delta(G) \leq \chi'(G) \leq \Delta(G)+1$.

O problema de coloração também pode ser formulado como um problema de decisão. Dado um grafo G e um número k, que pode ser fixo ou fazer parte da entrada do problema, pergunta-se se G admite uma k-coloração.

Existem alguns limites conhecidos para $\chi(G)$. O primeiro limite que vale trivialmente é $\chi(G) \leq |V(G)|$, mas alguns outros, mais interessantes, são citados a seguir:

- $\chi(G) \leq \Delta(G)+1$, que pode ser provado via coloração gulosa;
- $\chi(G) \leq \Delta(G)$ se G não for um grafo completo ou um ciclo ímpar (Teorema de Brooks);
- $\chi(G) \geq |V(G)|/\alpha(G)$;
- $\chi(G) \geq \omega(G)$.

Esses resultados podem ser encontrados em (WEST, 2000). Embora tenhamos alguma estimativa para o valor de $\chi(G)$, três problemas surgem. O primeiro é que esses limites dependem de parâmetros difíceis de calcular (encontrar $\alpha(G)$ ou $\omega(G)$ são problemas NP-difíceis). O segundo é que esses limites fornecem estimativas que podem ser muito ruins. Por exemplo, em (WEST, 2000) é possível encontrar uma construção devida a Mycielski, que produz uma família de grafos sem triângulos (consequentemente, com $\omega(G) = 2$) e com número cromático tão grande quanto se queira. O terceiro é que, sem informações adicionais sobre o grafo, ter um limite superior ou inferior para o número cromático não reduz, em geral, a classe de complexidade da solução. O problema da coloração para $k \geq 3$ fixo é NP-completo; uma demonstração é feita para $k = 3$ neste documento.

Esses resultados podem ser encontrados em (WEST, 2000). Embora tenhamos alguma estimativa para o valor de $\chi(G)$, três problemas surgem. O primeiro é que esses limites dependem de parâmetros difíceis de calcular (encontrar $\alpha(G)$ ou $\omega(G)$ são problemas NP-difíceis). O segundo é que esses limites fornecem estimativas que podem ser muito ruins. Por exemplo, em (WEST, 2000) é possível encontrar uma construção devida a Mycielski, que produz uma família de grafos sem triângulos (consequentemente, com $\omega(G) = 2$) e com número cromático tão grande quanto se queira. O terceiro é que, sem informações adicionais sobre o grafo, ter um limite superior ou inferior para o número cromático não reduz, em geral, a classe de complexidade da solução. O problema da coloração para $k \geq 3$ fixo é NP-completo; uma demonstração é feita para $k = 3$ neste documento.

3.2 A coloração como problema NP-completo

A prova de NP-completude de um problema Π é feita em dois passos. O primeiro é provar que $\Pi \in NP$. O segundo é escolher um problema sabidamente NP-completo e mostrar que ele admite uma redução polinomial no problema que se quer provar que é NP-completo (ou seja, mostrar que o problema é NP-difícil). O fato de a 3-coloração pertencer a NP é simples: dados um grafo G e uma coloração f, basta verificar se em todos os pares de vértices vale: $f(v_1) \neq f(v_2)$ ou $v_1v_2 \notin E(G)$. Ao todo são feitas $\binom{n}{2}$ comparações, onde $n = |V(G)|$, claramente polinomial no tamanho do grafo.

Em 1971, Stephen Cook provou que o *problema de satisfatibilidade* (SAT) é NP-completo. Definimos este problema mais formalmente na seção seguinte.

3.3 Forma normal conjuntiva e o 3SAT

O problema consiste em saber se é possível atribuir valores a *literais* (fórmulas atômicas ou negação de fórmulas atômicas) em uma expressão lógica de modo que a expressão seja verdadeira. Por exemplo, $x \vee y$ pode ser satisfeita, mas $x \wedge \neg x$ não pode. Diz-se que uma expressão lógica E está expressa em *forma normal conjuntiva* se satisfaz às seguintes regras:

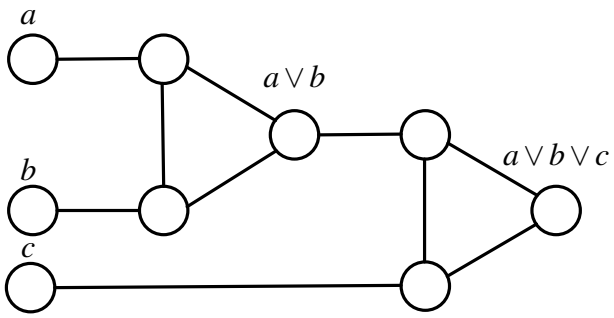
- E é uma conjunção de *cláusulas* $C_1, C_2, \ldots, C_m$;
- cada cláusula é uma disjunção de literais

No problema 3SAT, supõe-se que cada cláusula possui 3 literais. Dizemos que uma instância I do 3SAT com cláusulas $C_1, \ldots, C_m$ e literais $x_1, \ldots, x_n$ é *satisfatível* se existe uma função $f : \{x_1, \ldots, x_n\} \to \{V, F\}$ tal que I é verdadeira.

3.4 Forma normal conjuntiva e o 3SAT

Seja $C = (a \vee b \vee c)$ uma cláusula com literais a, b e c. A Figura 1 mostra um grafo produzido a partir de C. Suponha que as cores dos vértices pertençam ao conjunto $\{V, F, B\}$, representando respectivamente os valores verdadeiro, falso e uma cor auxiliar B. Note que se a e b forem falsos, o vértice $a \vee b$ deve receber necessariamente a cor F. Se, além disso, c for falso, o vértice $a \vee b \vee c$, dito *vértice de saída do dispositivo "ou"*, necessariamente receberá a cor F para que a coloração do dispositivo "ou" seja própria. Por outro lado, se pelo menos um dos literais for verdadeiro, existe uma coloração própria tal que o vértice $a \vee b \vee c$ recebe a cor V.

Figura 1 – Dispositivo “ou”



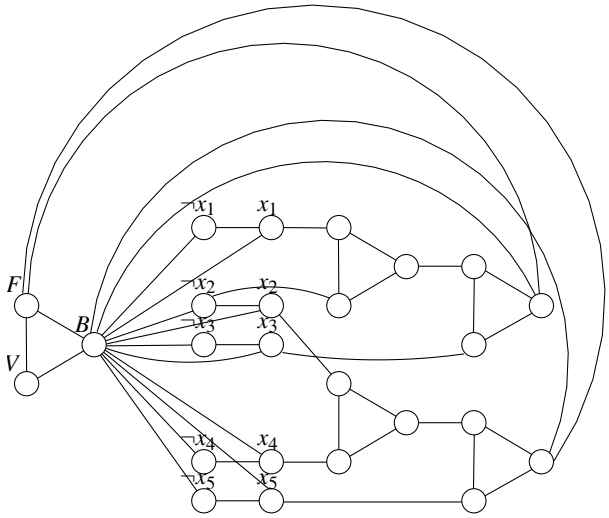
A demonstração a seguir foi proposta originalmente em (JOHNSON D.; GAREY; STOCKMEYER, 1976); a interpretação do grafo da Figura 1 como um circuito lógico é sugerida na seção de problemas do capítulo 34 de (CORMEN T.; LEISERSON, 2009). Dada uma instância L do 3SAT com cláusulas $C_1, \ldots, C_m$ e literais $x_1, \ldots, x_n$, busca-se um grafo G, cujos vértices e arestas sejam limitados por um polinômio em m e n, tal que G admite 3-coloração se, e somente se, L pode ser satisfeita.

Para aplicar o resultado acima a uma instância L do 3SAT, basta representar os literais, os dispositivos “ou” correspondentes a cada cláusula e alguma estrutura auxiliar para garantir que o grafo final admite 3-coloração se, e somente se, L pode ser satisfeita. Seja L uma instância do 3SAT com cláusulas $C_1, \ldots, C_m$ e literais $x_1, \ldots, x_n$. O grafo G correspondente é formado da seguinte maneira:

- Adicione um K_3 com os vértices V, F e B (estes irão representar as cores);
- para cada variável x_i, adicione dois vértices, chamados x_i e $\neg x_i$, ligados entre si e ao vértice de cor B do primeiro item;
- para cada cláusula C_j um dispositivo “ou” como o da Figura 1, com os vértices a, b e c identificados aos respectivos literais e o vértice de saída $a \vee b \vee c$ ligado aos vértices de cores B e F do primeiro item.

A Figura 2 mostra o grafo construído para a expressão $L = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee x_5)$.

Figura 2 – Grafo obtido para a instância $L = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee x_5)$



Proposição 3.4.1 *L pode ser satisfeita* $\iff$ *G admite 3-coloração.*

Demonstração: ($\Rightarrow$) Se L pode ser satisfeita, cada cláusula é verdadeira. E em cada cláusula, pelo menos um literal vai ser verdadeiro. Como visto acima, existe uma coloração para o dispositivo "ou" correspondente tal que o vértice de saída recebe a cor V. A coloração assim obtida é própria.

($\Leftarrow$) Se G é 3-colorível, então todos os vértices de saída dos dispositivos "ou" são coloridos com V (por construção, pois são vizinhos de vértices de cores B e F). Os literais são todos vizinhos de um vértice com a cor B, logo possuem cor V ou F. Um literal será verdadeiro ou falso conforme a cor recebida. Tome $C_j = (a \vee b \vee c)$ uma cláusula qualquer. Se os três literais fossem falsos, o vértice de saída teria a cor F, o que não ocorre. Logo, pelo menos um literal é verdadeiro em cada cláusula, e L é satisfeita.

□

Note que, para uma expressão L com m cláusulas e n literais, o grafo obtido possui $2n+6m+3$ vértices e $3n+12m+3$ arestas. Portanto, a transformação entre os dois problemas é polinomial.

3.5 Algoritmos e recursividade

Um *algoritmo* ou *procedimento* é uma sequência finita de operações sobre um objeto que produz um outro objeto como resultado, de forma semelhante a uma máquina de Turing mas sem necessariamente resolver um problema de decisão. Assim como funções, algoritmos podem ser definidos de maneira recursiva. Segue abaixo um algoritmo que calcula o n-ésimo número de Fibonacci, para $n \in \mathbb{N}$.

Algoritmo 1: Procedimento `Fibonacci(`n`)`

```
1 Procedimento Fibonacci(n):
2     se n > 2 então retorna Fibonacci(n − 1) + Fibonacci(n − 2)
3
4     senão se n = 2 ou n = 1 então retorna 1
5
6     senão retorna NULL
7
8
9 fim
```

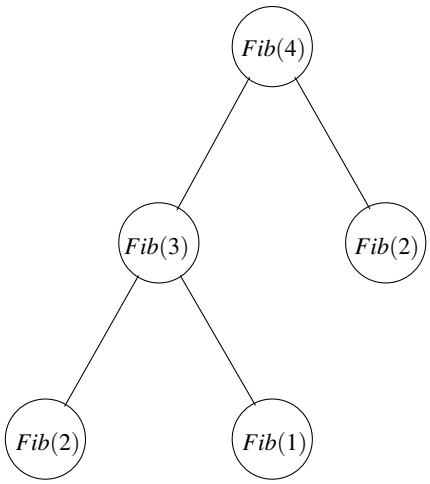
Dados um algoritmo recursivo $\mathscr{A}$ e um valor de entrada x, podemos definir uma *árvore de recursão* associada da seguinte maneira: Um primeiro vértice da árvore corresponde à execução de $\mathscr{A}(x)$. Cada vez que o algoritmo $\mathscr{A}$ faz referência a si mesmo para o cálculo de $\mathscr{A}(x)$, criamos um novo vértice, adjacente ao vértice da execução de $\mathscr{A}$ que o originou. Para o algoritmo mencionado acima, a árvore de recursão resultante para o cálculo do quarto número de Fibonacci é apresentada na Figura 3.

Dizemos que um algoritmo *resolve* um problema de decisão $L \subseteq \Sigma^*$ se, ao receber uma instância $I \in \Sigma^*$, retorna `SIM` quando $I \in L$ e `NÃO`, caso contrário. Algumas vezes, o algoritmo poderá produzir também uma testemunha para a resposta `SIM`.

3.6 Problemas FPT

Como mencionado no primeiro capítulo, decidir $\chi(G) \leq k$ para grafos livres de P_t pode ser resolvido em tempo polinomial, para alguns valores de k e t fixos. Observe porém que existe uma diferença importante entre um algoritmo com complexidade $O(n^{kt})$ e um de complexidade $O(2^{kt}n)$, sendo que este último, quando k e t são fixos, é visto como um algoritmo linear. Estas diferenças são capturadas na definição a seguir, encontrada em (CYGAN M.; FOMIN,

Figura 3 – Cálculo de $Fibonacci(4)$



2015).

Dado um alfabeto Σ, definimos como um *problema parametrizado* uma linguagem $L \subseteq \Sigma^* \times \mathbb{N}$. Para cada instância $I = (x,k) \in \Sigma^* \times \mathbb{N}$, dizemos que k é o parâmetro. Um problema parametrizado é dito *tratável a parâmetro fixo* ou FPT[1] quando pode ser resolvido em um tempo limitado por $O(f(k) \cdot n^{O(1)})$ para alguma função computável $f : \mathbb{N} \to \mathbb{N}$. Note que o expoente em n não pode depender de k. Esta classe de algoritmos tem sido bastante estudada nos últimos anos; um exemplo de tal algoritmo é explorado no Capítulo 5.

[1] o acrônimo vem do nome em inglês

4 ESTADO DA ARTE

Como mencionamos no capítulo introdutório, a coloração é um dos problemas mais difíceis dos 21 problemas de Karp, e que mesmo a aproximação do número cromático com um fator $n^{(1-\varepsilon)}$ é NP-difícil (ZUCKERMAN, D., 2006). Ao tentarmos investigar a coloração restrita à classe dos grafos livres de H, deparamo-nos com um resultado devido a Král' et al (KRÁL; KRATOCHVÍL; WOEGINGER, G., 2001), que afirma que o problema se torna polinomial quando H é um subgrafo induzido de P_4 ou de $P_3 + K_1$, e é NP-completo caso contrário. O passo seguinte é considerar o número de cores k fixado e não mais como parte dos dados de entrada do problema.

A motivação para o estudo da coloração de grafos livres de caminhos curtos se mostra clara a partir de duas constatações: a primeira é a de que o problema da k-coloração de grafos livres de H é NP-completo sempre que H possuir um ciclo, pois o problema é NP-completo sobre os conjuntos de grafos com cintura maior do que ou igual a g, para todo $g \geq 3$ (VADIM V. L.; KAMINSKI M., 2007); a segunda é a de que H não pode possuir um $K_{1,3}$ induzido, pois o conjunto dos grafos livres de $K_{1,3}$ é subconjunto dos grafos linha, para os quais o problema da k-coloração é NP-completo (HOLYER, 1981).

Portanto, se desejamos encontrar uma restrição do problema que possa ser resolvida em tempo polinomial proibindo grafos H induzidos, temos que H não pode conter ciclos nem vértices com grau maior do que 2. Ou seja, H é uma floresta linear. Se nos restringirmos aos casos em que H é conexo, chegamos ao problema da coloração de grafos livres de caminhos curtos. O estado atual do problema está resumido na Tabela 2. Os casos que ainda estão em aberto são $k = 3$ e $t \geq 8$; e $k = 4$ e $t = 6$. Note que todo grafo livre de P_t também é livre de $P_{t'}$, para todo $t' > t$. Logo, um resultado polinomial para uma combinação (k,t) pode ser extendida a valores $t' < t$; por outro lado, resultados de NP-completude podem ser extendidos a valores $t' > t$.

Tabela 2 – Complexidade da k-coloração de grafos livres de P_t

	$k = 3$	$k = 4$	$k = 5$	$k \geq 6$
$t \leq 5$	P [1]	P [1]	P [1]	P [1]
$t = 6$	P [2]	?	NP-c [3]	NP-c [3]
$t = 7$	P [4], [5]	NP-c [3]	NP-c	NP-c [6]
$t \geq 8$	?	NP-c	NP-c	NP-c

[1] (HOÀNG C.T.; KAMIŃSKI, 2010)

[2] (RANDERATH B. ; SCHIERMEYER, 2004)

[3] (HUANG, S., 2016)

[4] (CHUDNOVSKY M.; MACELI, 2014)

[5] (CHUDNOVSKY M.; MACELI, 2015)

[6] (BROERSMA H.; FOMIN, 2013)

Muitos dos resultados obtidos são frutos de publicações recentes. Pelo menos parte deles faz uso de técnicas voltadas para a k-coloração por listas, como (HOÀNG C.T.; KAMIŃSKI, 2010) e (HUANG, S., 2016), do qual a k-coloração é um caso particular (basta fazer todas as listas iguais a $\{1,\ldots,k\}$).Há avanços recentes para o caso em que $k = 4$ e $t = 6$.

Em (HUANG, S., 2016) Shenwei Huang conjectura que este caso admite solução polinomial e mostra que isso é verdadeiro no caso particular em que o grafo G também é livre de banner [1]. O mesmo artigo mostra que os casos em que $k = 4$ e $t = 7$; e $k = 5$, $t = 6$ são NP-completos através da existência de um tipo de grafo específico dito bem comportado, que não existe para $k = 4$ e $t = 6$. Um outro caso particular onde $k = 4$ e $t = 6$ que admite solução em tempo polinomial é o dos grafos livres de P_6 e C_5 (CHUDNOVSKY M.; MACELI, 2017).

Tanto em (HUANG, S., 2016) quanto em (CHUDNOVSKY M.; MACELI, 2017), adota-se uma estratégia de decompor o grafo em um conjunto de instâncias menores e servindo-se de propriedades estruturais das classes, uma vez que essas classes são hereditárias. Mencionamos que, à época da conclusão da escrita deste texto, foi anunciada uma solução para o caso em que $k = 4$ e $t = 6$, dividida em dois artigos, da autoria de Maria Chudnovsky, Sophie Spirkl e Mingxian Zhong (CHUDNOVSKY M.; SPIRKL, 2018a)(CHUDNOVSKY M.; SPIRKL, 2018b). Como tais artigos ainda não passaram por uma revisão por parte da comunidade acadêmica, achamos melhor não anunciar este caso como resolvido.

[1] Um *banner* é um grafo obtido a partir de um C_4, adicionando-se um vértice extra e tornando-o adjacente a exatamente um vértice do ciclo

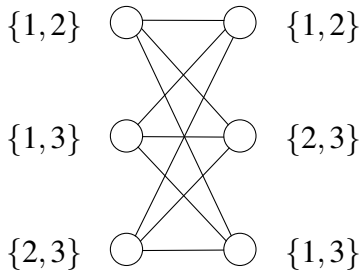
O caso em que $k = 3$ e $t \geq 8$ é endereçado em uma pré-publicação de autoria de Maria Chudnovsky, Jan Goedgebeur, Oliver Schaudt e Mingxian Zhong (CHUDNOVSKY M.; GOEDGEBEUR, 2017). A abordagem é baseada no conceito de obstrução. Chamamos um par $(G, \mathscr{L})$, com $\mathscr{L}(v) \subseteq \{1,2,3\}$ para todo $v \in V(G)$, uma *obstrução minimal por listas* se não existe $\mathscr{L}$-coloração de G mas o par $(A, \mathscr{L})$ é colorível para todo subgrafo A de G próprio. O artigo em questão explora propriedades estruturais das obstruções minimais por listas. Embora não resolva o problema, o artigo mostra dois resultados. O primeiro afirma que o número de grafos 4-críticos livres de H se e somente se H é um subgrafo induzido de P_6, $2P_3$ ou $P_4 + kP_1$ para algum $k \in \mathbb{N}$. O segundo afirma que existe um número finito de obstruções por listas $(G, \mathscr{L})$ com G livre de H se e somente se H é um subgrafo induzido de P_6 ou $P_4 + kP_1$ para algum $k \in \mathbb{N}$.

5 O PROBLEMA DA ESCOLHA

Este capítulo destina-se a mostrar a estratégia usada em (GOLOVACH P.;HEGGERNES, 2009) para demonstrar que o problema da escolha restrito a grafos livres de P_5 é FPT quando parametrizado por k. As provas que envolvem resultados polinomiais costumam ser construtivas, isto é, dadas explicitamente por algoritmos que resolvem o problema em tempo polinomial.

Um exemplo clássico de grafo 2-colorível que não é 2-escolhível é o $K_{3,3}$, exibido na Figura 4. Mais geralmente, é possível mostrar que mesmo para algumas classes de grafos, como os bipartidos (para os quais vale $\chi(G) = 2$), a diferença entre o número de escolha e o número cromático pode ser arbitrariamente grande. Em particular, se $m = \binom{2k-1}{k}$, então $K_{m,m}$ não é k-escolhível (ERDŐS P.; RUBIN; TAYLOR, 1979).

Figura 4 – O grafo $K_{3,3}$ não é 2-escolhível



Teorema 5.0.1 (ERDŐS P.; RUBIN; TAYLOR, 1979) *Se $m = \binom{2k-1}{k}$, então $K_{m,m}$ não é k-escolhível.*

Demonstração:

Note que $\binom{2k-1}{k}$ é o número de subconjuntos com k elementos de um conjunto com $2k-1$ elementos. Suponha que $K_{m,m}$ seja k-escolhível e chamemos os dois conjuntos independentes de $K_{m,m}$ de A e B. Em particular, $K_{m,m}$ admite coloração por listas se as listas correspondentes aos vértices de A forem os subconjuntos de $\{1,\ldots,2k-1\}$ com k elementos (e o mesmo para B). Como, para todo $L \subseteq \{1,\ldots,2k-1\}$ tal que $|L| \leq k-1$, existe $v \in A$ tal que $L(v) \cap L = \emptyset$, temos que pelo menos k cores são usadas em A. Mas k destas cores correspondem exatamente à paleta de um dos vértices de B. Logo, $K_{m,m}$ não admite coloração própria para as listas atribuídas e não pode ser k-escolhível.

□

5.1 Definições para o problema

Precisamos de algumas definições adicionais para apresentarmos o algoritmo. Sejam G um grafo, $W = \{w_1, \ldots, w_r\} \subseteq V(G)$, $H = G - W$ e $\mathscr{L}$ uma k-atribuição sobre W. Definimos L como o conjunto das cores da atribuição $\mathscr{L}$; ℓ a maior cor em L; $\mathbb{L}$ o produto cartesiano das listas (cada $c \in \mathbb{L}$ é uma escolha de cores para os $w_i \in W$ dentro das listas de cada vértice); $\mathbb{X}$ um conjunto de escolhas de cores para os $w_i \in W$ dentro da atribuição $\mathscr{L}$. Mais formalmente:

$$L = \bigcup_{i=1}^{r} \mathscr{L}(w_i)$$
$$\ell = \max L$$
$$\mathbb{L} = \mathscr{L}(w_1) \times \ldots \times \mathscr{L}(w_r)$$
$$\mathbb{X} = 2^{\mathbb{L}}$$

De forma análoga, podemos definir L', ℓ', $\mathbb{L}'$ e $\mathbb{X}'$ a partir de uma atribuição $\mathscr{L}'$ sobre $V(H)$. Dados $c = (c_1, \ldots, c_r) \in \mathbb{L}$ e $\mathscr{L}'$ uma k-atribuição sobre H, dizemos que *c concorda com $\mathscr{L}'$* se existe uma $\mathscr{L}'$-coloração f de H tal que a coloração $(f + c) : V(G) \to [\max(\ell, \ell')]$ é própria, onde

$$(f+c)(v) = \begin{cases} f(v), & \text{se } v \in V(H) \\ c_i, & \text{se } v = w_i \end{cases} \tag{5.1}$$

Dado $X \in \mathbb{X}$, dizemos que *X concorda com $\mathscr{L}'$* se c concorda com $\mathscr{L}'$ para todo $c \in X$.

5.2 O Algoritmo

A prova de existência de solução polinomial para um problema normalmente se dá por construção explícita de um algoritmo. Nesta seção mostramos que o algoritmo exibido em (GOLOVACH P.;HEGGERNES, 2009) é correto e possui tempo de solução polinomial. O algoritmo se baseia em dois teoremas, apresentados a seguir. O primeiro é encontrado em (ALON, 1993) e o segundo em (BACSÓ G.; TUZA, 1990).

Teorema 5.2.1 (ALON, 1993) *Sejam G um grafo e s um inteiro positivo. Se o grau médio de G satisfaz $\overline{d}(G) > 4\binom{s^4}{s} log(2\binom{s^4}{s})$, então $ch(G) > s$.*

Teorema 5.2.2 (BACSÓ G.; TUZA, 1990) *Todo grafo conexo G livre de P_5 possui um conjunto dominante que é uma clique ou um P_3. Além disso, um tal conjunto pode ser encontrado em tempo polinomial.*

Antes de apresentar o algoritmo que resolve o problema da escolha (ou seja, dado um grafo G, determinar se este é k-escolhível), vamos definir um problema de decisão auxiliar, que tem como entradas um grafo G livre de P_5, um subconjunto W de $V(G)$ e uma k-atribuição $\mathscr{L}$ sobre W.

`Problema da Concordância`

Entrada: $(G, W, \mathscr{L})$

Pergunta: Existe $\mathscr{X} = \{X_1, \ldots, X_s\} \subseteq \mathbb{X}$, tal que, para toda k-atribuição $\mathscr{L}'$ sobre $H = G - W$, existe $X_i \in \mathscr{X}$ que concorda com $\mathscr{L}'$?

O problema da concordância é resolvido pelo algoritmo recursivo `Colorir`, apresentado no pseudocódigo abaixo, que retorna uma testemunha $\mathscr{X}$ para o problema de decisão. Para simplificar a afirmação de que $\mathscr{X}$ é testemunha para a instância $(G, W, \mathscr{L})$ do problema de concordância, diremos que $\mathscr{X}$ *resolve* $(G, W, \mathscr{L})$. Os símbolos $\emptyset$ e $\{\emptyset\}$ são distintos nesse contexto. O primeiro representa uma família vazia de fato e corresponde a uma resposta negativa para o problema de decisão; o segundo corresponde a uma resposta positiva para o caso $(G, \emptyset, \emptyset)$, conforme discutido a seguir.

Vejamos o que acontece quando tentamos resolver o problema da concordância para $(G, \emptyset, \emptyset)$. Neste caso, temos $W = \emptyset \subseteq V(G)$, $H = G - W = G$, $\mathscr{L} = \emptyset$, $\mathbb{L} = \emptyset$, $\mathbb{X} = \{\emptyset\}$ e podemos convencionar $\ell = 0$. Observando a Equação 5.1, constatamos que $X = \emptyset \in \mathbb{X}$ é compatível com uma k-atribuição $\mathscr{L}'$ sobre H quando existe uma $\mathscr{L}'$-coloração própria sobre G (já que não existe nenhum $w_i \in W$ e $V(H) = V(G)$). Reciprocamente, se G é k-colorível para toda k-atribuição $\mathscr{L}'$, então $\emptyset$ concorda com $\mathscr{L}'$, ou seja, $\{\emptyset\}$ é a resposta para o problema da concordância para $(G, \emptyset, \emptyset)$. Em outras palavras, resolver o problema da concordância para $(G, \emptyset, \emptyset)$ é equivalente a resolver o problema da escolha sobre G.

Por outro lado, se a resposta para o problema da concordância é positiva e $W \neq \emptyset$, então a família $\mathscr{X}$ é necessariamente não-vazia. Se $\mathscr{X}$ fosse vazio, existiria uma k-atribuição $\mathscr{L}'$ sobre $V(H)$ com a qual nenhuma $\mathscr{L}$-coloração de W concorda, ou seja, $\mathscr{L}'' = \mathscr{L} \cup \mathscr{L}'$ seria uma k-atribuição de cores sobre $V(G)$ que não admite coloração própria, o que implicaria

$ch(G) > k$.

Um ponto importante deve ser feito a respeito do valor máximo do conjunto de cores usado nas atribuições $\mathscr{L}'$ definidas sobre H para uma instância $(G, W, \mathscr{L})$ dada. Queremos saber se existe $\mathscr{X}$ tal que, para toda k-atribuição $\mathscr{L}' : V(H) \to \binom{\mathbb{N}}{k}$, temos que existe $X \in \mathscr{X}$ que concorda com $\mathscr{L}'$. O que podemos observar é que é possível concluir algo a este respeito sem precisar testar infinitos conjuntos. A título de ilustração, consideremos o problema da concordância sobre o grafo G definido por $V(G) = \{v_1, v_2\}$, $E(G) = \{v_1v_2\}$, para $k = 2$, $W = \{v_1\}$, $\mathscr{L}(v_1) = \{1,2\}$. É suficiente testar os casos em que $\mathscr{L}'(v_2)$ é igual a $\{1,2\}$, $\{1,3\}$, $\{1,4\}$, $\{2,3\}$, $\{2,4\}$ e $\{3,4\}$, pois todos os casos de interseção estão inclusos, assim como o caso mais extremo, em que as listas são disjuntas. Fazer $\mathscr{L}'(v_2)$ igual a $\{3,4\}$ ou $\{4,5\}$ não faz diferença para fins de interseções de listas de coloração. Eis o que justifica o limite da linha 16 do pseudocódigo sobre as listas de cores.

Assumimos, para efeito de cálculo de complexidade e prova de corretude, que G é conexo, pois o número de escolha de um grafo G é o máximo dos números de escolha de suas componentes conexas. Assumimos também que $k \geq 3$, pois o problema da escolha para $k = 2$ pode ser resolvido em tempo polinomial para grafos quaisquer (ERDŐS P.; RUBIN; TAYLOR, 1979).

Em linhas gerais, o algoritmo aplicado a uma instância $(G, W, \mathscr{L})$ funciona tomando-se $H = G - W$ e $U \subseteq V(H)$ um conjunto dominante de H, obtido por aplicação do Teorema 5.2.2. Dado que $k \geq 3$, se $|U| > k$ então U é clique e, portanto, $ch(G) \geq \chi(G) \geq |U| > k$ e o algoritmo nos retorna uma família vazia de conjuntos, que corresponde a uma resposta negativa para o problema de decisão. Caso o conjunto dominante tenha passado por esse crivo e H passe pelo teste do grau médio (linha 14), o algoritmo analisa todas as k-atribuições $\mathscr{L}'$ relevantes de U, testando-as à exaustão se $U = V(H)$ ou fazendo uma chamada recursiva a `Colorir`$(G, W \cup U, \mathscr{L} \cup \mathscr{L}')$ caso contrário. Os detalhes do pseudocódigo do algoritmo são explorados na prova de corretude.

Algoritmo 2: Procedimento `Colorir(`G`,` W`,` $\mathscr{L}$`)`

```
1  Procedimento Colorir(G, W, 𝓛):
2      D ← 4·C(k⁴,k)·log(2·C(k⁴,k))
3      H ← G − W
4      L ← 𝓛(w_1) ∪ ... ∪ 𝓛(w_r)
5      se W = ∅ então ℓ ← 0
6      senão ℓ ← max L
7      𝕃 ← 𝓛(w_1) × ... × 𝓛(w_r)
8      𝕏 ← 2^𝕃
9      Encontre U = {u_1,...,u_p} conjunto dominante de H, com G[U] clique
         ou P_3
10     p ← |U|
11     𝒳 ← ∅
12     se p > k então retorna ∅
13
14     se d̄(H) > D então retorna ∅
15
16     para todo 𝓛' : U → [ℓ + kp] faça
17         se U = V(H) então
18             se W ≠ ∅ então
19                 X ← ∅
20                 para todo 𝓛'-Coloração f de H faça
21                     X ← X ∪ {c ∈ 𝕃; (f + c) é própria}
22                 fim
23                 se X ≠ ∅ então
24                     𝒳 ← 𝒳 ∪ {X}
25                 senão
26                     retorna ∅
27                 fim
28             senão
29                 se ∄ 𝓛'-col. própria então retorna ∅
30             fim
31         senão
32             Encontre H_1,...,H_q componentes conexas de H − U
33             para i ∈ {1,...,q} faça
34                 F_i ← G[W ∪ U ∪ V(H_i)]
35                 𝒳_i ← Colorir(F_i, W ∪ U, 𝓛 ∪ 𝓛')
36                 se 𝒳_i = ∅ então retorna ∅
37             fim
38             𝒴 ← 𝒳_1
39             para i ∈ {2,...,q} faça
40                 𝒵 ← ∅
41                 para todo X ∈ 𝒳_i, Y ∈ 𝒴 faça
42                     se X ∩ Y ≠ ∅ então
43                         𝒵 ← 𝒵 ∪ (X ∩ Y)
44                     senão
45                         retorna ∅
46                     fim
47                     𝒴 ← 𝒵
48                 fim
49             fim
50             para todo Z ∈ 𝒵 faça
51                 se W ≠ ∅ então
52                     X ← {(c(w_1),...,c(w_r)) : c ∈ Z, c(w_i) ≠ c(u_j) se w_iu_j ∈
                         E(G) e c(u_i) ≠ c(u_j) se u_iu_j ∈ E(G)}
53                     se X ≠ ∅ então
54                         𝒳 ← 𝒳 ∪ X
55                     senão
56                         retorna ∅
57                     fim
58                 senão
59                     X ← {c ∈ Z : c(u_i) ≠ c(u_j) se u_iu_j ∈ E(G)}
60                     se X ≠ ∅ então 𝒳 ← 𝒳 ∪ X
61                     senão retorna ∅
62                 fim
63             fim
64         fim
65     fim
66     se W = ∅ então 𝒳 ← {∅}
67     retorna 𝒳
68 fim
```

5.3 Prova da Corretude

A prova da corretude do algoritmo é feita por indução em $|V(G)|$. Observe que todo subgrafo induzido de um grafo livre de P_5 é livre de P_5. Portanto, isso não será mencionado na prova, pois ficará implícito.

Teorema 5.3.1 *Seja $\mathscr{X} = (X_1, \ldots, X_s)$ a família de elementos de $\mathbb{X}$ produzida por $\texttt{Colorir}(G, W, \mathscr{L})$. Então, $\mathscr{X}$ resolve $(G, W, \mathscr{L})$. Reciprocamente, se existe $\mathscr{X} \neq \emptyset$ que resolve $(G, W, \mathscr{L})$, então $\texttt{Colorir}(G, W, \mathscr{L})$ retorna uma família não-vazia.*

Demonstração: Tome uma instância $(G, W, \mathscr{L})$ do problema de concordância e seja $\mathscr{X} = \texttt{Colorir}(G, W, \mathscr{L})$. Provamos que, se $\mathscr{X} \neq \emptyset$, então $\mathscr{X}$ resolve $(G, W, \mathscr{L})$ e que, se $\mathscr{X} = \emptyset$, então não existe $\mathscr{X}'$ que resolve $(G, W, \mathscr{L})$. Primeiro suponha que $\mathscr{X} \neq \emptyset$. Analisemos o algoritmo para saber como $\mathscr{X}$ é gerado. Na linha 9, produz-se um conjunto dominante U, que é uma clique ou um P_3, em tempo polinomial, o que é possível dado que G é livre de P_5 pelo Teorema 5.2.2. As linhas 12 e 14, embora não sejam suficientes para resolver o problema, rejeitam instâncias com base em condições necessárias e desempenham um papel importante para o controle da complexidade. Note que se $|U| > k$, então U é uma clique (pois $k \geq 3$ e a desigualdade $|U| > k$ é estrita). Neste caso, $ch(G) \geq ch(G[U]) \geq \chi(G[U]) > k$ e o algoritmo pára na linha 12. Se $\overline{d}(H) > D$, o algoritmo pára na linha 14; observe que $ch(G) \geq ch(H) > k$ pelo Teorema 5.2.1. Temos então dois casos a considerar:

I) $U = V(H)$: este caso se desmembra em dois.

   a) Se $W = \emptyset$, para cada k-atribuição de listas $\mathscr{L}'$ sobre U (linha 16), testa-se por exaustão (ou seja, todas as possibilidades são verificadas) se existe $\mathscr{L}'$-coloração sobre $G[U]$ (linhas 28 a 30). Mas $U = V(H) = V(G)$, pois $W = \emptyset$. Como por hipótese $\mathscr{X} \neq \emptyset$, isso significa que o algoritmo termina na linha 67. Por conta da linha 66, temos $\mathscr{X} = \{\emptyset\}$. Por hipótese, o algoritmo passa pelo laço da linha 16, pelo ramo correspondente ao valor verdadeiro para a condição da linha 17 (linhas 18 a 30) e pelo ramo correspondente ao valor falso para a condição da linha 18 (linha 29). Se a condição da linha 29 fosse verdadeira em algum momento, o algoritmo retornaria $\mathscr{X} = \emptyset$, mas $\mathscr{X} = \{\emptyset\} \neq \emptyset$. Logo, temos que toda atribuição $\mathscr{L}'$ definida no laço da linha 16 admite coloração própria, e $\{\emptyset\}$ resolve $(G, W, \mathscr{L})$.

   b) Se $W \neq \emptyset$, o algoritmo passa pelo laço da linha 16, pelo ramo correspondente ao valor verdadeiro para a condição da linha 17 e pelo ramo correspondente ao valor

verdadeiro para a condição da linha 18 (linhas 19 a 27). Assim, para cada $\mathscr{L}'$-coloração de H (linha 20), adicionamos ao conjunto $\mathscr{X}$ um subconjunto de $\mathbb{L}$ que concorda com $\mathscr{L}'$ (linha 21). Por hipótese, o algoritmo retorna $\mathscr{X}$ não-vazio, o que significa que a condição da linha 23 é sempre verdadeira; caso contrário o algoritmo terminaria na linha 26. Ou seja, o conjunto $\mathscr{X}$ é tal que, para toda $\mathscr{L}'$-atribuição de H definida na linha 16, existe $X \in \mathscr{X}$ que concorda com $\mathscr{L}'$. Em outras palavras, $\mathscr{X}$ resolve $(G,W,\mathscr{L})$.

II) $U \neq V(H)$: para cada k-atribuição de listas $\mathscr{L}'$ sobre U definida na linha 16, o algoritmo segue os passos descritos entre as linhas 31 e 64. Sejam $H_1,\ldots,H_q$ as componentes conexas de $H-U$. Para cada $i \in \{1,\ldots,q\}$ definimos $F_i = G[W \cup U \cup V(H_i)]$ e fazemos a chamada recursiva $\mathscr{X}_i = \texttt{Colorir}(F_i, W \cup U, \mathscr{L} \cup \mathscr{L}')$. Como $\mathscr{X}$ é não-vazio, a condição da linha 36 é sempre falsa. Por hipótese de indução, $\mathscr{X}_i$ resolve $(F_i, W \cup U, \mathscr{L} \cup \mathscr{L}')$ para todo $i \in \{1,\ldots,q\}$. Nas linhas 38 a 49, obtemos uma família $\mathscr{Z}$ tal que todo $Z \in 2^{\mathbb{L}\times\mathbb{L}'}$ em $\mathscr{Z}$ (caso $W = \emptyset$, temos $Z \in 2^{\mathbb{L}'}$) concorda com todos os H_i. Por conta da linha 43, as colorações em $\mathscr{Z}$ dão as mesmas cores sobre $W \cup U$ para cada H_i.

O caso em que $W = \emptyset$ é tratado nas linhas 59 a 61. A condição da linha 60 é sempre verdadeira; caso contrário o algoritmo avançaria para a linha 61 e teríamos $\mathscr{X}$ vazio. Logo, para toda atribuição $\mathscr{L}'$ sobre U definida no laço da linha 16 temos uma $\mathscr{L}'$-coloração de U que concorda com toda atribuição toda k-atribuição $\mathscr{L}''$ sobre H_i, pela construção de $\mathscr{Z}$. Como os conjuntos H_i são disjuntos, então a $\mathscr{L}'$-coloração de U em questão concorda com $H-U$, ou seja, G é k-escolhível. Como visto acima, isso é equivalente a dizer que $\{\emptyset\}$ resolve $(G,\emptyset,\emptyset)$. De fato, o algoritmo sai do laço da linha 50 esgotando todos os elementos de $\mathscr{Z}$, sai do laço da linha 16 esgotando todas as listas $\mathscr{L}'$ e segue para a linha 66, retornando $\{\emptyset\}$ em 67.

O caso em que $W \neq \emptyset$ é tratado nas linhas 52 a 57. A condição da linha 52 verifica se é possível adicionar à família $\mathscr{X}$ uma restrição de Z a W para cada $Z \in \mathscr{Z}$. Note que c concorda com todos os H_i, e a condição garante que X concorde com U. Como a condição da linha 53 é sempre verdadeira, por hipótese, cada X é não vazio. Assim, W concorda com $G[U] \cup H_1 \cup \ldots \cup H_q = H$, ou seja, todos os elementos de $\mathscr{X}$ concordam com H para qualquer k-atribuição $\mathscr{L}'$ sobre H. Em outras palavras, $\mathscr{X}$ resolve $(G,W,\mathscr{L})$.

Agora, suponha que $\mathscr{X} = \emptyset$. Queremos mostrar que não existe $\mathscr{X}'$ que resolve $(G,W,\mathscr{L})$. Em particular, o algoritmo pára em uma das linhas seguintes: 12, 14, 26, 29, 36, 45,

56 ou 61. Analisemos cada um desses casos.

I) Linha 12: se $|U| = p > k \geq 3$, então U é uma clique (não pode ser P_3 pois a desigualdade é estrita) de tamanho p. Logo, $\chi(H) \geq \chi(U) > k$. Neste caso, a atribuição $\mathscr{L}'$ sobre $V(H)$ definida por $\mathscr{L}'(v) = \{1,\ldots,k\}$ para todo $v \in V(H)$ não admite $\mathscr{L}'$-coloração própria e, portanto, não existe c que concorda com $\mathscr{L}'$.

II) Linha 14: observe que, pelo Teorema 5.2.1, H não é k-escolhível, ou seja, existe k-atribuição $\mathscr{L}'$ sobre H para a qual H não é k-colorível. Logo, nenhum $c \in \mathbb{L}$ pode concordar com $\mathscr{L}'$, o que implica que não existe $\mathscr{X}'$ que resolve $(G,W,\mathscr{L})$.

III) Linha 26: esta linha é atingida se $X = \emptyset$ na linha 23, o que por sua vez ocorre se toda coloração $(f+c) : V(G) \to \mathbb{N}$ onde c é $\mathscr{L}$-coloração de W e f é $\mathscr{L}'$-coloração de H não é própria para alguma atribuição $\mathscr{L}'$ sobre H. Logo, não existe $c \in \mathbb{L}$ que concorda com $\mathscr{L}'$, o que implica que não existe $\mathscr{X}'$ que resolve $(G,W,\mathscr{L})$.

IV) Linha 29: se não existe $\mathscr{L}'$-coloração para H para alguma k-atribuição $\mathscr{L}'$, então H não é k-escolhível. Logo, não existe $\mathscr{X}'$ que resolve $(G,W,\mathscr{L})$.

V) Linha 36: dadas as atribuições $\mathscr{L}$ sobre W e $\mathscr{L}'$ sobre U, existe alguma atribuição $\mathscr{L}''$ sobre $V(H_i)$ com a qual nenhuma $(\mathscr{L} \cup \mathscr{L}')$-coloração de $W \cup U$ concorda. Vê-se então que nenhuma $\mathscr{L}$-coloração de W concorda com $\mathscr{L}' \cup \mathscr{L}''$ e que isso pode ser facilmente estendido para H. Logo, não existe $\mathscr{X}'$ que resolve $(G,W,\mathscr{L})$.

VI) Linha 45: seja $i \in \{2,\ldots,q\}$ a iteração do laço da linha 39 na qual o algoritmo parou. No processo iterativo compreendido entre as linhas 38 e 49, sabemos que $\mathscr{X}$ resolve $(F_i, W \cup U, \mathscr{L} \cup \mathscr{L}')$ e $\mathscr{Y}$ resolve $(F_j, W \cup U, \mathscr{L} \cup \mathscr{L}')$, para $1 \leq j < i$. Sejam $X \in 2^{\mathbb{L} \times \mathbb{L}'}$ (ou $2^{\mathbb{L}'}$ caso $W = \emptyset$) o conjunto das $(\mathscr{L} \cup \mathscr{L}')$-colorações de $W \cup U$ que concordam com todas as $\mathscr{L}''_1$-atribuições sobre H_i e $Y \in 2^{\mathbb{L} \times \mathbb{L}'}$ (ou $2^{\mathbb{L}'}$ caso $W = \emptyset$) o conjunto das $(\mathscr{L} \cup \mathscr{L}')$-colorações de $W \cup U$ que concordam com todas as $\mathscr{L}''_2$-atribuições sobre H_j, para todo $j \in \{1,\ldots,i-1\}$. Se $X \cap Y = \emptyset$, então existem atribuições $\mathscr{L}''_1$ sobre H_i e $\mathscr{L}''_2$ sobre H_j para algum j tais que nenhuma $(\mathscr{L} \cup \mathscr{L}')$-coloração de $U \cup W$ concorda com a atribuição $\mathscr{L}''_1 \cup \mathscr{L}''_2$ sobre $H_i \cup H_j$. Isso pode ser facilmente estendido para H, logo não existe $\mathscr{X}'$ que resolve $(G,W,\mathscr{L})$.

VII) Linha 56: neste caso, o conjunto X na linha 52 é vazio (condição da linha 53). Sabemos que $\mathscr{Z}$ resolve $(F_i, W \cup U, \mathscr{L} \cup \mathscr{L}')$ para todo i, o que implica em $\mathscr{Z}$ resolve $(G, W \cup U, \mathscr{L} \cup \mathscr{L}')$ para uma atribuição $\mathscr{L}'$ dada sobre U. Se X é vazio significa que existe uma atribuição $\mathscr{L}''$ sobre $V(H) - U$ tal que nenhuma $\mathscr{L} \cup \mathscr{L}'$-coloração de $W \cup U$ concorda

com $\mathscr{L}''$, para alguma atribuição $\mathscr{L}'$ sobre U. Defina $\tilde{\mathscr{L}}$ sobre H de tal modo que $\tilde{\mathscr{L}}|_U = \mathscr{L}'$ e $\tilde{\mathscr{L}}|_{V(H)-U} = \mathscr{L}''$. Se existisse $\mathscr{X}'$ que resolvesse $(G,W,\mathscr{L})$, bastaria tomar uma $\mathscr{L}$-coloração c e $f \in \tilde{\mathscr{L}}$ tal que $f+c$ seja própria sobre G. Temos então uma $\mathscr{L}\cup\mathscr{L}'$-coloração de $W\cup U$ que concorda com $\mathscr{L}''$. Absurdo, pois X é vazio. Logo, não existe $\mathscr{X}'$ que resolve $(G,W,\mathscr{L})$.

VIII) Linha 61: neste caso, o conjunto X na linha 59 é vazio, para alguma $\mathscr{L}'$-atribuição sobre U. Ou seja, $G[W\cup U]$ não é k-escolhível, logo não existe $\mathscr{X}'$ que resolve $(G,W,\mathscr{L})$.

IX) Linha 67: note que se o algoritmo não parou em nenhuma das linhas mencionadas acima, adicionamos pelo menos um conjunto não-vazio a $\mathscr{X}$. A adição é feita nas linhas 24 ou 54, caso $W \neq \emptyset$, ou na linha 66 caso contrário.

□

5.4 Complexidade do algoritmo

Nesta seção provamos o resultado seguinte:

Teorema 5.4.1 *O algoritmo apresentado é FPT, com parâmetro k.*

Para o cálculo de tempo de execução do algoritmo, vamos encontrar limitantes para o número de chamadas do procedimento e para o número de operações dentro de cada chamada.

O número de chamadas do procedimento é limitado pela profundidade da árvore de recursão vezes o número de folhas. Este último é no máximo $|V(G)| = n$. Observe que chamadas são feitas somente na linha 35. Será útil, portanto, limitar primeiramente a quantidade de conjuntos dominantes distintos para os quais alguma chamada é feita. Sejam $U_1,\ldots,U_h$ tais conjuntos.

Como $|U_i| \leq k$, então $|U_1\cup\ldots\cup U_h| \leq hk$. Seja $F = G[U_1\cup\ldots\cup U_h]$. Como U_i é conjunto dominante de $U_{i+1},\ldots,U_h$, temos $\sum_{v\in U_i} d_F(v) \geq h-1$. Assim, para h conjuntos, temos $\sum_{v\in V(F)} d(v) \geq h(h-1)$. Da definição do grau médio, temos $\overline{d}(F) = \frac{1}{|V(F)|}\sum_{v\in V(F)} d(v)$. Como $\sum_{v\in V(F)} d(v) \geq h(h-1)$ e $|V(F)| = |U_1\cup\ldots\cup U_h| \leq hk$, então temos que $\overline{d}(F) \geq \frac{h(h-1)}{hk} = \frac{h-1}{k}$. Note que, se $h > kD+1$, então $\overline{d}(F) > \frac{(kD+1)-1}{k} = D$ e o procedimento é interrompido; logo a profundidade h da árvore de recursão é no máximo $kD+1$. Substituindo o valor de D e usando a desigualdade $\binom{a}{b} \leq 2^a$, temos:

$$kD+1 = 4k\binom{k^4}{k}\log(2\binom{k^4}{k})+1 \leq 4k\cdot 2^{k^4}(k^4+1)+1 = O(k^5\cdot 2^{k^4})$$

Multiplicando pelo número de folhas, que é no máximo n, temos $O(k^5 \cdot 2^{k^4} \cdot n)$ chamadas para o algoritmo.

Resta estimar o tempo de execução de cada rodada do procedimento. O conjunto dominante U pode ser obtido em tempo polinomial, pelos resultados de (BACSÓ G.; TUZA, 1990). Sabemos também que se $|U| > k$ o algoritmo termina sem mais operações além da própria verificação do tamanho do conjunto. Podemos supor para a estimativa que $|U| \leq k$. Como a árvore de recursão possui profundidade h, as listas de cores são escolhidas dentro do conjunto $\{1,\ldots,h \cdot k^2\}$. Segue a explicação deste fato: note que na linha 16 do algoritmo as cores são limitadas por ℓ, k e p. Na primeira chamada, `Colorir(G,∅,∅)`, temos $\ell = 0$, $h = 1$ e o primeiro conjunto para escolha de listas é $\{1,\ldots,k \cdot p\} \subseteq \{1,\ldots,k^2\}$, pois $p \leq k$ (caso contrário o procedimento termina antes do laço). Suponha que as listas estejam contidas em $\{1,\ldots,(h-1) \cdot k^2\}$ para uma árvore de recursão de profundidade $(h-1)$. Tome uma chamada recursiva a partir de qualquer vértice da árvore na profundidade $(h-1)$. Temos, por hipótese de indução, que $\ell \leq (h-1) \cdot k^2$. No laço da linha 16, as novas listas são tomadas em $[\ell + k \cdot p]$. Mas

$$\ell + k \cdot p \leq (h-1) \cdot k^2 + k \cdot p \leq (h-1) \cdot k^2 + k^2 = h \cdot k^2$$

Logo, para uma profundidade de recursão h, as listas de cores são tomadas dentro do conjunto $\{1,\ldots,h \cdot k^2\}$.

Sabemos também que $h \leq kD+1$, pelo parágrafo anterior; assim, todas as listas de cores são subconjuntos com k elementos do conjunto $\{1,\ldots,(kD+1)k^2\}$. Existem $\binom{(kD+1)k^2}{k}$ possíveis listas e, portanto, existem no máximo $\binom{(kD+1)k^2}{k}^k$ possibilidades de atribuição de listas aos vértices de U. Do parágrafo anterior, sabemos que $kD+1 = O(k^5 \cdot 2^{k^4})$. Combinando este fato com o de que $\binom{a}{b} \leq 2^a$, temos no máximo $(2^{k^2 O(k^5 \cdot 2^{k^4})})^k = 2^{O(k^8 \cdot 2^{k^4})}$ possibilidades de atribuição de listas aos vértices de U.

Uma vez limitado o número de listas possíveis, devemos olhar para o tamanho do conjunto gerado. A saída do algoritmo é um conjunto vazio, caso o grafo G não seja k-escolhível, ou uma família $\mathscr{X} = \{X_1,\ldots,X_s\} \subseteq \mathbb{X}$, caso G o seja. Com uma profundidade de recursão h e conjuntos dominantes com tamanho máximo k (exceto quando o algoritmo termina nas verificações iniciais), o tamanho de W deve ser no máximo $k(kD+1)$. Logo, a lista que é retornada caso G seja k-escolhível possui no máximo $2^{k(kD+1)} = 2^{O(k^6 2^{k^4})}$ conjuntos, novamente usando a estimativa de $kD+1$ do parágrafo anterior. Usando esses limitantes mais o fato de que o número de componentes conexas q de $H-U$ nos casos em que $U \neq V(H)$ satisfaz $q \leq n$, podemos estimar que o número de operações em cada chamada do procedimento é no máximo

$2^{O(k^8 2^{k^4})} \cdot n^c$ para alguma constante c. Levando em conta o número de chamadas, podemos limitar o tempo de execução do procedimento `Colorir` por $2^{O(k^8 2^{k^4})} \cdot n^s$ para alguma constante s. Fica assim demonstrado que o algoritmo é FPT.

6 NP - COMPLETUDE

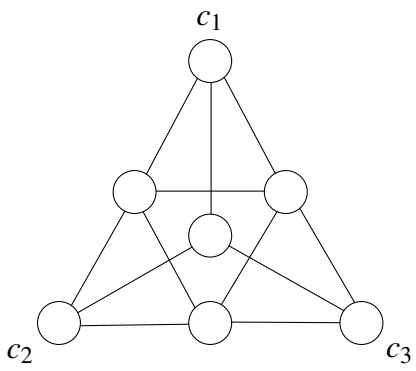
O propósito deste capítulo é mostrar a estratégia usada para provar a NP-completude do problema de k-coloração de grafos livres de P_t, nos casos $(k = 5, t = 6)$ e $(k = 4, t = 7)$ em (HUANG, S., 2016). Conforme visto na Seção 3.4, o fato de que o problema da k-coloração (e suas restrições a classes de grafos) pertence a NP é de fácil verificação: para verificar se uma coloração de um grafo G é de fato uma coloração própria, basta testar se as extremidades de cada aresta possuem cores distintas, um procedimento que é quadrático no número de vértices. Desta forma, resta mostrar que uma restrição em particular do problema é NP-difícil. Faremos uso de uma construção similar à da Seção 3.4. Embora um pouco mais sofisticada, conta com elementos similares. A demonstração se dá em três passos:

- Toma-se um grafo H, dito *bem comportado*, em um sentido preciso a ser definido abaixo.
- Transforma-se uma instância I qualquer do 3SAT em uma instância equivalente $G_{H,I}$ do problema de k-coloração. Esta construção faz uso da instância I e do grafo H.
- Prova-se que se um grafo bem comportado H em questão é livre de P_t para algum inteiro t, o grafo construído $G_{H,I}$ também o é.

Assim, o problema da k-coloração de grafos livres de P_t é NP-completo sempre que encontrarmos um grafo H bem comportado livre de P_t. A partir desse ponto, basta encontrar os grafos convenientes para k e t especificados.

Um grafo k-crítico H é dito *bem comportado* quando contém três vértices independentes c_1, c_2 e c_3 tais que $\omega(H - \{c_1, c_2, c_3\}) = \omega(H) = k - 1$. A Figura 5 mostra um exemplo para $k = 4$.

Figura 5 – Um grafo 4-crítico bem comportado

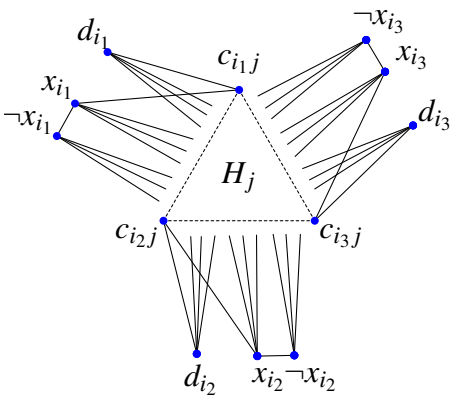


Dados uma instância I do 3SAT com variáveis $X = \{x_1, x_2, \ldots, x_n\}$ e cláusulas $\mathscr{C} = \{C_1, C_2, \ldots, C_m\}$ e um grafo k-crítico bem comportado H, o grafo $G_{H,I}$ é construído como segue:

- Para cada variável x_i, uma *componente literal* T_i, com dois vértices adjacentes x_i e $\neg x_i$, que serão ditos vértices do tipo X.
- Ainda para cada variável x_i, um vértice d_i, que será do tipo D.
- Para cada cláusula $C_j = (y_{i_1} \vee y_{i_2} \vee y_{i_3})$[1], uma cópia H_j de H. Os três vértices independentes de H_j[2] serão denotados $c_{i_1 j}, c_{i_2 j}, c_{i_3 j}$ e serão ditos vértices do tipo C. Os demais vértices de H_j serão ditos do tipo U.
- Para cada vértice c_{ij}, chamaremos x_i ou $\neg x_i$ o *vértice literal correspondente* a c_{ij} conforme x_i ou $\neg x_i$ pertença a C_j.
- Cada vértice do tipo U é adjacente a todos os vértices dos tipos D e X.
- Cada vértice c_{ij} do tipo C é adjacente a d_i e a seu literal correspondente.

O procedimento é ilustrado na Figura 6.

Figura 6 – Construção de $G_{H,I}$ dados H e $C_j = (x_{i_1} \vee x_{i_2} \vee x_{i_3})$



Lema 6.0.1 *Sejam I instância do 3SAT e H k-crítico bem comportado. Seja $G_{H,I}$ obtido como descrito acima. Temos que I é satisfatível se, e somente se, $G_{H,I}$ é $(k+1)$-colorível.*

Demonstração: ($\Rightarrow$) Considere uma atribuição $\sigma : X \to \{V, F\}$ que satisfaz I. Iremos construir uma $(k+1)$-coloração própria $\phi : V(G_{H,I}) \to \{1, 2, \ldots, k+1\}$ de $G_{H,I}$, começando pelos vértices

[1] y_{i_1} é x_{i_1} ou $\neg x_{i_1}$; os outros dois literais são análogos

[2] convenientemente escolhidos em função do fato de H_j ser bem comportado

dos tipos D e X. De início, temos $\phi(d_i) = k+1$ para todo $d_i \in D$. Se $\sigma(x_i) = V$, então $\phi(x_i) = k+1$ e $\phi(\neg x_i) = k$. Caso contrário, $\phi(x_i) = k$ e $\phi(\neg x_i) = k+1$.

Como I é verdadeira, cada cláusula $C_j = (y_{i_1} \vee y_{i_2} \vee y_{i_3})$ possui pelo menos um literal verdadeiro y_{i_t}, para algum $t \in \{1,2,3\}$. Note que vizinhos de $c_{i_t j}$ (d_{i_t} e seu literal correspondente) receberam a cor $k+1$. Assim, se $\phi(c_{i_t j}) = k$, a coloração ϕ continua própria. Note que H_j é k-crítico, logo $H_j - c_{i_t j}$ admite uma coloração com $k-1$ cores. Assim, podemos tomar uma coloração qualquer de H_j usando as cores de 1 a $k-1$. Finalmente, note que o fato de os vértices de tipo U serem vizinhos de todos os vértices de D e X não causa problema, pois $\phi(U) \subseteq \{1,2,\ldots,k-1\}$ e $\phi(X \cup D) = \{k, k+1\}$. Logo, a coloração assim obtida é própria.

($\Leftarrow$) Considere agora ϕ uma $(k+1)$-coloração própria de $G_{H,I}$. Iremos construir uma atribuição σ que satisfaz I. Tome H_i uma das cópias de H criadas na construção de $G_{H,I}$, $i \in \{1,2,\ldots,m\}$. Por definição, sabemos que $\omega(H_j \cap U) = k-1$. Tome R_i clique maximal em $H_i \cap U$. Temos que $R = R_i \cup \{x_i, \neg x_i\}$ é uma clique de tamanho $k+1$, e como ϕ é própria, todos os vértices de R recebem cores diferentes. Sem perda de generalidade, podemos supor que $\{\phi(x_i), \phi(\neg x_i)\} = \{k, k+1\}$. Segue imediatamente que $\phi(R_i) = \{1,2,\ldots,k-1\}$. Note que disso decorre que $\phi(\{x_j, \neg x_j\}) = \{k, k+1\}$ para todo j, pois $R_i \cup \{x_j, \neg x_j\}$ é clique de tamanho $k+1$, para todo $j \in \{1,\ldots,n\}$, e $\phi(d_i) \in \{k, k+1\}$, pois R_i é adjacente a todos os vértices do tipo D. Segue ainda que $\phi(u) \in \{1,2,\ldots,k-1\}$ para todo vértice u do tipo U, pois U é completo a X e as cores k e $k+1$ já aparecem em X.

Definimos $\sigma(x_i) = V$ se $\phi(x_i) = \phi(d_i)$ e $\sigma(x_i) = F$ caso contrário. Argumentamos a seguir que σ assim definida sobre os literais satisfaz I.

Suponhamos que não seja o caso. Então, pelo menos uma cláusula $C_j = (y_{i_1} \vee y_{i_2} \vee y_{i_3})$ é falsa. Para cada $t \in \{1,2,3\}$, por um abuso de linguagem, denote também por y_{i_t} o vértice literal correspondente a y_{i_t}, ou seja, y_{i_t} será x_{i_t} ou $\neg x_{i_t}$. Mas como X e D usam apenas as cores k e $k+1$, temos então que $\{\phi(y_{i_t}), \phi(d_{i_t})\} = \{k, k+1\}$, para cada $t \in \{1,2,3\}$.Como $c_{i_t j}$ é adjacente a y_{i_t} e d_{i_t}, temos $\phi(c_{i_t j}) < k$, para todo $t \in \{1,2,3\}$. Pelo parágrafo anterior, temos então que H_j está colorido com cores inferiores a k. Absurdo, pois $\chi(H) = k$.

□

O resultado acima mostra que a $(k+1)$-coloração é um problema NP-completo sempre que existir um grafo k-crítico bem comportado H.

Lema 6.0.2 *Sejam H um grafo k-crítico bem comportado, $t \geq 6$ inteiro e I uma instância qualquer do 3SAT. Se H é livre de P_t, então $G_{H,I}$ também o é.*

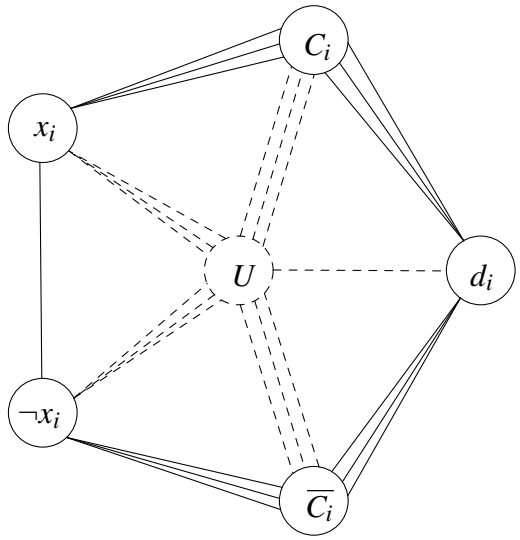
Demonstração: Por contradição, suponha que P é um caminho induzido de tamanho t em $G_{H,I}$. Temos então três propriedades:

(a) O caminho P possui pelo menos um vértice em $X \cup D$. Note que se P não intersecta $X \cup D$, então P está contido em alguma cópia de H, pois $G - (X \cup D)$ é uma união disjunta de cópias de H e P é conexo. Mas H é livre de P_t, absurdo.

(b) O mesmo caminho possui pelo menos um vértice do tipo U. Para a prova desta propriedade precisamos do conceito de substituição. Dados dois grafos G_1 e G_2, *substituir* G_1 em $v \in V(G_2)$ ou substituir $v \in V(G_2)$ por G_1 significa retirar v de G_2, incluir uma cópia de G_1 e ligar todos os vértices dessa cópia a $N_{G_2}(v)$. Quando não houver ambiguidade, a menção ao vértice substituído pode ser omitida.

Seja C_i o conjunto dos vértices do tipo C adjacentes a x_i, e seja $\overline{C_i}$ o conjunto dos vértices do tipo C adjacentes a $\neg x_i$. Cada componente conexa de $G_{H,I} - U$ possui uma estrutura proveniente da substituição dos vértices de um C_5 por cinco conjuntos. Os cinco conjuntos, em ordem (cíclica) são: $A_1 = \{x_i\}$, $A_2 = C_i$, $A_3 = \{d_i\}$, $A_4 = \overline{C_i}$ e $A_5 = \{\neg x_i\}$. O leitor poderá se convencer comparando o processo de construção de $G_{H,I}$ com a Figura 7.

Figura 7 – Ilustração de uma componente conexa de $G_{H,I} - U$



Como C_5 é livre de P_t para $t \geq 5$ e cada um dos A_j é um conjunto independente, as componentes conexas de $G_{H,I} - U$ são livres de P_t. Logo, P possui pelo menos um vértice do tipo U.

(c) O caminho P possui no máximo três vértices em $X \cup D \cup U$ e eles são consecutivos em P. Pelos itens (a) e (b), temos que P possui pelo menos um vértice $u \in U$ e um vértice $x \in (X \cup D)$. Como cada vértice de U é adjacente a todos os vértices de D e de X, eles são consecutivos em P. Sejam x^+ vizinho de x diferente de u (existe no máximo um, pois P é caminho) e u^+ vizinho de u diferente de x (pelo mesmo argumento existe no máximo um). É imediato que $x^+ \notin (X \cup D)$, pois teríamos um triângulo induzido, contrariando o fato de que P é caminho induzido. Da mesma forma, $u^+ \notin U$. Mais ainda, pelo menos um dos vértices x^+ ou u^+ está em C, caso contrário teríamos $x^+ \in U$ e $u^+ \in X \cup D$, e eles seriam consecutivos, violando o fato de que P é caminho induzido. De forma análoga, os demais vértices de P estão em C. Assim, $X \cup D \cup U$ possui no máximo três vértices de P (x, u e um dos vértices x^+ e u^+) e eles são consecutivos.

Segue da propriedade (c) que no máximo três vértices consecutivos de P não pertencem a C. Como $t \geq 6$, C possui vértices consecutivos de P. Absurdo, pois C é conjunto independente. Logo, $G_{H,I}$ é livre de P_t.

□

Temos então dois resultados importantes. O primeiro diz que podemos transformar de uma maneira polinomial o 3SAT em um problema de $(k+1)$-coloração desde que exista um grafo k-crítico bem comportado. O segundo diz que se o grafo k-crítico bem comportado em questão é livre de P_t, os grafos gerados no problema de coloração também são. Ora, se um problema restrito a um conjunto $\mathscr{A}$ é NP-completo, também será NP-completo se restrito a todo superconjunto de $\mathscr{A}$. Assim, o resultado principal desta seção segue de maneira imediata.

Teorema 6.0.1 (HUANG, S., 2016) *Sejam $t \geq 6$ e $k \geq 3$ inteiros. Se existe um grafo H k-crítico bem comportado que é livre de P_t, então o problema de $(k+1)$-coloração em grafos livres de P_t é NP-completo.*

O trabalho que resta é mostrar que existem grafos k-críticos bem comportados livres de P_t para os casos em que se deseja provar a NP-completude. O grafo da Figura 5 é 4-crítico, bem comportado e livre de P_6. Logo, a 5-coloração de grafos livres de P_6 é um problema NP-completo. Da mesma forma, C_7 é 3-crítico, bem comportado e livre de P_7. O fato de que C_7 é 3-crítico é de verificação imediata: retirando-se qualquer vértice temos um P_6, e $\chi(P_6) = 2$. Também

é imediato que C_7 é livre de P_7, pois o único subgrafo induzido de C_7 com 7 vértices possui uma aresta a mais do que P_7. Para verificar que C_7 é bem comportado, podemos enumerar seus vértices em sequência: $\{c_1, \ldots, c_7\}$. Sabemos que $\omega(C_7) = 2$, pois C_7 é livre de triângulos. Os pontos $\{c_1, c_3, c_5\}$ formam um conjunto independente e $\omega(C_7 - \{c_1, c_3, c_5\}) = \omega(P_2 + 2P_1) = 2$. Logo, a 4-coloração de grafos livres de P_7 também é um problema NP-completo.

7 CONCLUSÃO

Neste trabalho foi abordado o problema da coloração do ponto de vista de complexidade. Citamos o resultado presente em (KRÁL; KRATOCHVÍL; WOEGINGER, G., 2001) sobre a coloração de grafos livres de H e comentamos sobre as consequências dos resultados de (VADIM V. L.; KAMINSKI M., 2007) e (HOLYER, 1981) sobre a natureza de H no problema da k-coloração para um valor de k fixado, concluindo que as possíveis classes de grafos livres de H ocorrem quando H é uma floresta linear. Apresentamos um resumo dos resultados conhecidos quando H é conexo na Tabela 1 e comentamos sobre os trabalhos recentes para os casos em que $k = 4$ e $t = 6$; e $k = 3$ e $t \geq 8$, os únicos que restam em aberto dentro desta premissa. Apresentamos no Capítulo 5 e uma prova de um caso polinomial, por meio de apresentação explícita de um algoritmo que resolve o problema. Apresentamos no Capítulo 6 uma prova de NP-completude através de redução do 3SAT.

Sobre o caso em aberto em que $k = 4$ e $t = 6$, mencionamos que Shenwei Huang conjectura em (HUANG, S., 2016) que o caso em que $k = 4$ e $t = 6$ admite solução polinomial e prova que isso é válido no caso particular em que o grafo também é livre de banner. Ainda em (HUANG, S., 2016), prova-se que não existe grafo 3-crítico bem comportado livre de P_6, o que faz que o Teorema 6.0.1 não possa ser aplicado para este caso. Porém, a não existência destes grafos não implica que o problema seja polinomial, uma vez que o Teorema 6.0.1 funciona somente em uma direção. Mencionamos ainda que existe um trabalho da autoria de Maria Chudnovsky *et al.* (CHUDNOVSKY M.; SPIRKL, 2018a)(CHUDNOVSKY M.; SPIRKL, 2018b), atualmente sob revisão, que classifica o caso como polinomial via prova construtiva. O algoritmo se baseia na extensão de pré-colorações por listas com propriedades especiais, denotadas pré-colorações excelentes

O caso em que $k = 3$ e ≥ 8, que continua em aberto, é explorado por Maria Chudnovsky *et al.* em uma pré-publicação (CHUDNOVSKY M.; GOEDGEBEUR, 2017) que, embora não resolva o problema, define um tipo de grafo chamado obstrução como uma extensão de grafo crítico a colorações por listas e aponta para quais classes grafos H o número obstruções livres de H é finito.

Uma situação não explorada aqui é o caso dos grafos livres de dois ou mais grafos $H_1, \ldots, H_r$. Ainda que a Tabela 1 seja preenchida, os casos em que H não é conexo estão longe de ser resolvidos e está ainda longe de existir uma família $\mathscr{H}$ que resolva uma dicotomia do tipo: seja G um grafo livre de H e k um inteiro fixo, decidir se $\chi(G) \leq k$ pode ser feito em tempo polinomial se $H \in \mathscr{H}$ e é NP-completo caso contrário.

REFERÊNCIAS

AARDAL K.; HOESEL, V. S. . K. A. . M. C. . S. A. Models and solution techniques for frequency assignment problems. **Annals of Operations Research**, New York, v. 153, n. 1, p. 79–129, 2007.

ALON, N. Restricted colorings of graphs. **Surveys in combinatorics**, United States, v. 187, p. 1–33, 1993.

APPEL K.; HAKEN, W. Every planar map is four colorable. i. discharging. **Illinois Journal of Mathematics**, United States, v. 21, p. 429–490, 1977.

APPEL K.; HAKEN, W. Every planar map is four colorable. ii. reducibility. **Illinois Journal of Mathematics**, United States, v. 21, p. 429–490, 1977.

BACSÓ G.; TUZA, Z. Dominating cliques in p_5-free graphs. **Periodica Mathematica Hungarica**, Hungary, v. 21, p. 303–308, 1990.

BROERSMA H.; FOMIN, F. V. . G. P. A. P. D. Three complexity results on coloring pk-free graphs. **European journal of combinatorics**, London, v. 34, n. 3, p. 609–619, 2013.

CAYLEY. On the colouring of maps. London, v. 1, n. 4, p. 259–261, 1879.

CHAITIN G. ; AUSLANDER, M. C. A. C. J. H. M. M. P. Register allocation via coloring. **Computer languages**, Netherlands, v. 6, n. 1, p. 47–57, 1981.

CHUDNOVSKY M.; GOEDGEBEUR, J. S. Z. M. Obstructions for three-coloring and list three-coloring h-free graphs. **arXiv preprint arXiv:1703.05684**, 2017.

CHUDNOVSKY M.; MACELI, P. S. J. Z. M. 4-coloring p6-free graphs with no induced 5-cycles. **Journal of Graph Theory**, United States, v. 84, n. 3, p. 262–285, 2017.

CHUDNOVSKY M.; MACELI, P. Z. M. Three-coloring graphs with no induced seven-vertex path i: the triangle-free case. **arXiv preprint arXiv:1409.5164**, United States, 2014.

CHUDNOVSKY M.; MACELI, P. Z. M. Three-coloring graphs with no induced seven-vertex path ii: using a triangle. **arXiv preprint arXiv:1503.03573**, United States, 2015.

CHUDNOVSKY M.; SPIRKL, S. Z. M. Four-coloring p_6-free graphs. i. extending an excellent precoloring. **arXiv preprint arXiv:1802.02282**, 2018.

CHUDNOVSKY M.; SPIRKL, S. Z. M. Four-coloring p_6-free graphs. ii. finding an excellent precoloring. **arXiv preprint arXiv:1802.02283**, 2018.

COOK, S. The complexity of theorem proving procedures. **Proceedings of the Third Annual ACM Symposium on Theory of Computing**, New York, v. 1, p. 151–158, 1971.

CORMEN T.; LEISERSON, C. R. R. S. C. **Introduction to Algorithms**Massachusetts: The MIT Press. 2009.

CYGAN M.; FOMIN, F. K. L. L. D. D. P. M. P. M. S. S. **Parameterized Algorithm**New York: Springer Publishing Company, Incorporated. 2015.

ERDŐS P.; RUBIN; TAYLOR, H. Choosability in graphs. **West Coast Conference on Combinatorics, Graph Theory and Computing**, 1979.

GAREY M. R. ; JOHNSON, D. S. **Computers and intractability; A Guide to the Theory of NP-Completeness** United States: W. H. Freeman. 1979.

GOLOVACH P.;HEGGERNES, P. Choosability of p5-free graphs. **MFCS '09 Proceedings of the 34th International Symposium on Mathematical Foundations of Computer Science**, Poland, v. 1, p. 382–391, 2009.

GROTSCHEL, M.; LOVASZ, L.; SCHRIJVER, A. Polynomial algorithms for perfect graphs. **Annals of Discrete Mathematics**, Netherlands, v. 21, p. 325–356, 1984.

HEAWOOD, P. Map-colour theorem. **Quarterly Journal of Mathematics**, Oxford, v. 24, n. 1, p. 332–339, 1890.

HOÀNG C.T.; KAMIŃSKI, M. . L. V. Deciding k-colorability of p_5-free graphs in polynomial time. **Algorithmica**, New York, v. 57, p. 74–81, 2010.

HOLYER, I. The np-completeness of edge-coloring. **SIAM Journal on Computing**, United States, v. 10, n. 4, p. 718–720, 1981.

HUANG, S. Improved complexity results on k-coloring p_t-free graphs. **European Journal of Combinatorics**, Netherlands, v. 51, p. 336–346, 2016.

JOHNSON D.; GAREY; STOCKMEYER, L. Some simplified np-complete graph problems. **Theoretical Computer Science**, Germany, v. 1, p. 237–267, 1976.

KARP, R. M. Reducibility among combinatorial problems. **Springer**, United States, v. 58, n. 1, p. 85–103, 1972.

KRÁL; KRATOCHVÍL; WOEGINGER, G. Complexity of coloring graphs without forbidden induced subgraphs. **Graph-Theoretic Concepts in Computer Science**, New York, v. 1, p. 254–262, 2001.

MACKENZIE, D. **Mechanizing proof: computing, risk, and trust** Massachusetts: MIT Press. 2004.

RANDERATH B. ; SCHIERMEYER, I. 3-colorability $\in$ p for p6-free graphs. **Discrete Applied Mathematics**, United States, v. 136, n. 2-3, p. 299–313, 2004.

ROBERTSON N.; SANDERS, D. S. P. T. R. The four-color theorem. **Journal of Combinatorial Theory, Series B**, Canada, v. 70, p. 2–44, 1977.

STOCKMEYER, L. Planar 3-colorability is polynomial complete. **ACM Sigact News**, United States, v. 5, n. 3, p. 19–25, 1973.

VADIM V. L.; KAMINSKI M. Coloring edges and vertices of graphs without short or long cycles. **Contributions to Discrete Mathematics**, Canada, v. 2, p. 336–346, 2007.

VIZING. On an estimate of the chromatic class of a p-graph. **Diskret. Analiz.**, v. 3, p. 25–30, 1964.

WEST, D. **Introduction to Graph Theory**London:Pearson. 2000.

ZUCKERMAN, D. Linear degree extractors and the inapproximability of max clique and chromatic number. **Annual ACM Symposium on Theory of Computing**, New York, v. 1, n. 10, p. 681–690, 2006.