



## PROBLEMAS DE CORTES DE ARESTAS MÁXIMOS E MÍNIMOS EM GRAFOS

Rubens André Sucupira

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadores: Sulamita Klein  
Luerbio Faria

Rio de Janeiro  
Janeiro de 2017

PROBLEMAS DE CORTES DE ARESTAS MÁXIMOS E MÍNIMOS EM  
GRAFOS

Rubens André Sucupira

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

---

Prof. Sulamita Klein, D.Sc.

---

Prof. Luerbio Faria, D.Sc.

---

Prof. Jayme Luiz Szwarcfiter, Ph.D.

---

Prof. Uéverton dos Santos Souza, D.Sc.

---

Prof. Ignasi Sau, Ph.D.

RIO DE JANEIRO, RJ – BRASIL  
JANEIRO DE 2017

Sucupira, Rubens André

Problemas de Cortes de Arestas Máximos e Mínimos  
em Grafos/Rubens André Sucupira. – Rio de Janeiro:  
UFRJ/COPPE, 2017.

XIII, 116 p.: il.; 29, 7cm.

Orientadores: Sulamita Klein

Luerbio Faria

Tese (doutorado) – UFRJ/COPPE/Programa de  
Engenharia de Sistemas e Computação, 2017.

Referências Bibliográficas: p. 108 – 113.

1. MAXCUT.
  2. Grafos Cordais.
  3. Grafos Split.
  4. Grafos aresta-coloridos.
  5. Algoritmos Aproximativos.
  6. Complexidade Parametrizada.
- I. Klein, Sulamita *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

*À minha avó Eny Rocha dos  
Santos e minha mãe Eliane  
Santos Sucupira.*

# Agradecimentos

Em primeiro lugar, agradeço a Deus porque sem Deus nada é possível. É Ele que nos dá inspiração e força para trilharmos os caminhos de nossas vidas.

Em segundo lugar, agradeço à minha mãe e minha avó (já falecida) que sempre me apoiaram e incentivaram nessa tarefa gratificante, mas ao mesmo tempo árdua, que foi a preparação de todo esse trabalho, desde as disciplinas cursadas até as pesquisas, artigos, viagens e a preparação desse texto. Minha avó sempre sonhou em me ver com o título de doutor e creio que nesse momento ela esteja muito feliz.

Agradeço também aos meus amigos do Lac, em especial a Luís Felipe Ignácio Cunha, Lucila Bento, Carlos Vinícius e Alexander Andrade de Melo que muito me ajudaram nos momentos de apuros computacionais e burocráticos, e meus irmãos acadêmicos Fernanda Couto, Diego Nicodemos e Sancrey Rodrigues Alves por todo o apoio prestado ao longo desse processo.

Aos meus orientadores Sulamita Klein e Luerbio Faria, sem os quais nada disso teria sido possível, agradeço imensamente. Em especial a Luerbio Faria que me apresentou a Sula e possibilitou o início de todo esse processo. Com eles aprendi muito e pude me tornar o pesquisador que sou hoje, aprimorando minhas técnicas de pesquisa, escrita de artigos e preparação de apresentações. A participação deles foi fundamental na obtenção de muitos dos resultados presentes nessa tese.

É imprescindível destacar aqui a ajuda de dois professores pesquisadores muito queridos: Ignasi Sau e Uéverton dos Santos. Além de serem pesquisadores brilhantes que muito colaboraram nesta pesquisa como coautores de dois artigos, são pessoas maravilhosas, humanas, admiráveis.

Ao professor Jayme Szwarcfiter que sempre tive como um exemplo a ser seguido, um modelo de conduta e de pesquisador, agradeço por ter feito parte da minha banca de qualificação e por ter aceito fazer parte da banca de defesa juntamente com os professores Ignasi Sau e Uéverton dos Santos.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

## PROBLEMAS DE CORTES DE ARESTAS MÁXIMOS E MÍNIMOS EM GRAFOS

Rubens André Sucupira

Janeiro/2017

Orientadores: Sulamita Klein

Luerbio Faria

Programa: Engenharia de Sistemas e Computação

Iniciamos este trabalho mostrando que o problema do Corte Máximo Simples (SIMPLE MAXCUT) em grafos fortemente cordais é NP-completo, o que resolve um problema em aberto da *Coluna Johnson* [47]. Determinamos em seguida um algoritmo  $\frac{2}{3}$ -aproximativo para este problema em grafos split e resolvemos o problema para a subclasse dos grafos  $(k, n)$ -split cheios, determinando exatamente um corte máximo simples. Partimos então para o estudo da complexidade parametrizada do problema: apoiados no PROBLEMA DA DETERMINAÇÃO DO SUBGRAFO BALANCEADO DE TAMANHO MÁXIMO EM UM GRAFO SINALIZADO que generaliza o problema do Corte Máximo Simples mostramos que este último problema é tratável por parâmetro fixo em grafos split e grafos  $(r, \ell)$  exibindo um núcleo quadrático em ambos os casos. Obtivemos também núcleo linear para a subclasse dos grafos  $d^*$ -split. Finalmente consideramos os problemas de cortes máximo e mínimo de arestas em um grafo aresta-colorido, mostramos que ambos são problemas NP-completos mesmo em grafos completos e são tratáveis por parâmetro fixo.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

## ON SIMPLE MAXCUT PROBLEM FOR CHORDAL GRAPHS

Rubens André Sucupira

January/2017

Advisors: Sulamita Klein

Luerbio Faria

Department: Systems Engineering and Computer Science

We start this work showing that SIMPLE MAXCUT restricted to strongly chordal graphs is NP-complete, posed as open on *Johnson's Column* [47]. Next we find a  $\frac{2}{3}$ -approximative algorithm to solve this problem for split graphs and we find an exact algorithm for full  $(k, n)$ -split graphs. As we already know that this problem is NP-complete for generic graphs, we study its parameterized complexity using the problem: given a signed graph  $G$  we want to find a balanced signed subgraph of  $G$  with maximum size. This problem is a generalization of SIMPLE MAXCUT. We prove that SIMPLE MAXCUT is fixed parameter tractable on split graphs and  $(r, \ell)$  graphs exhibiting a quadratic kernel in both cases. We also find a linear kernel for  $d^*$ -split graphs. At last we consider the problems of MAXCUT and MINCUT on edge colored graphs, we show that both problems are NP-complete even on complete graphs and these problems are FPT.

# Sumário

<b>Lista de Figuras</b>	<b>x</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Organização do texto . . . . .	2
1.2 Definições Básicas e Notações . . . . .	3
1.2.1 Complexidade Clássica . . . . .	5
1.2.2 Algoritmos Aproximativos . . . . .	7
1.2.3 Complexidade Parametrizada . . . . .	8
1.3 Partições de Vértices e Cortes de Arestas . . . . .	10
1.4 Limite inferior para o corte máximo simples . . . . .	12
1.5 MAXCUT é NP-completo . . . . .	14
1.6 Casos especiais . . . . .	14
1.7 O Trabalho de Karp . . . . .	16
1.8 O Problema do Corte Máximo Simples . . . . .	17
<b>2 Complexidade do Problema do Corte Máximo para Grafos Fortemente Cordais</b>	<b>23</b>
2.1 Grafos Cordais . . . . .	23
2.1.1 Caracterização e Reconhecimento . . . . .	23
2.2 Grafos Fortemente Cordais . . . . .	25
2.2.1 Caracterização e Reconhecimento . . . . .	27
2.3 SIMPLE MAX CUT em Grafos Fortemente Cordais . . . . .	29
2.3.1 Esquema de eliminação forte de $G'$ . . . . .	30
<b>3 Algoritmos Aproximativos Polinomiais para o Problema do Corte Máximo em Grafos Split</b>	<b>35</b>
3.1 Definições Preliminares . . . . .	35
3.2 Algoritmos Aproximativos Polinomiais . . . . .	36
3.2.1 Um Algoritmo $\frac{2}{3}$ -aproximativo para SIMPLE MAX CUT . . . . .	37
3.3 Grafos $(k, n)$ -Split Cheios . . . . .	39

<b>4 Complexidade Parametrizada do Problema do Corte Máximo Sinalizado</b>	<b>45</b>
4.1 Resultados Gerais . . . . .	48
4.2 Núcleos . . . . .	50
4.3 Núcleo Quadrático para Grafos $(r, \ell)$ . . . . .	53
4.3.1 Núcleo Quadrático em Grafos Split . . . . .	56
4.3.2 Generalização para Grafos $(r, \ell)$ . . . . .	60
4.4 Núcleo Linear em Subclasses de Grafos Split . . . . .	64
4.4.1 MAX CUT é NP-completo em Grafos $d^*$ -Split . . . . .	65
4.4.2 O Algoritmo de Kernelização . . . . .	67
<b>5 Cortes de Arestas em Grafos Aresta-Coloridos</b>	<b>71</b>
5.1 Introdução . . . . .	71
5.2 Preliminares . . . . .	74
5.3 Resultados de NP-completude para o corte colorido mínimo . . . . .	75
5.4 Resultados de complexidade parametrizada para o corte colorido mínimo . . . . .	80
5.5 Resultados de inaproximabilidade para o corte colorido mínimo . . . . .	85
5.6 Resultados polinomiais para o corte colorido mínimo . . . . .	87
5.7 Resultados de NP-completude para o corte colorido máximo . . . . .	91
5.8 Algoritmo 2-aproximativo para o corte colorido máximo . . . . .	101
5.9 Resultados de complexidade parametrizada para o corte colorido máximo . . . . .	101
5.10 Resultados polinomiais para o corte colorido máximo . . . . .	102
<b>6 Conclusões</b>	<b>104</b>
<b>Referências Bibliográficas</b>	<b>108</b>
<b>Índice Remissivo</b>	<b>114</b>

# Lista de Figuras

1.1	Um grafo $G$ com sua clique máxima e o grafo complementar $\overline{G}$ com seu conjunto independente máximo. . . . .	8
1.2	(a) Um grafo $G$ e (b) um corte de arestas de $G$ de custo 9. . . . .	11
1.3	Quatro casos possíveis para a disposição dos vértices $u$ e $v$ . Somente os casos (a) e (b) são favoráveis . . . . .	12
1.4	Grafo $G$ associado ao conjunto $C = \{(x_1, \overline{x_2}); (\overline{x_1}, x_2)\}$ . As arestas com linha cheia pertencem a $E_1$ , enquanto as pontilhadas formam o conjunto $E_2$ . . . . .	20
1.5	Partição correspondente à atribuição $x_1 = F, x_2 = F$ . . . . .	21
2.1	(a) Um grafo cordal (grafo de Hajós) e (b) um não cordal (roda $W_6$ ). . . . .	23
2.2	Um grafo cordal que possui um esquema de eliminação perfeita $(1, 4, 5, 8, 2, 7, 3, 6)$ . . . . .	24
2.3	(a) Ilustração do esquema de eliminação perfeita $(1, 4, 5, 8, 2, 7, 3, 6)$ do grafo da Figura 2.2: (b),(c),(d),(e),(f),(g). . . . .	25
2.4	(a) Grafo não cordal com uma corda ímpar e (b) Grafo fortemente cordal. . . . .	26
2.5	(a) Um 3-sol completo (grafo de Hajós), (b) um 4-sol completo e (c) um 4-sol incompleto. . . . .	26
2.6	condição (i). . . . .	27
2.7	condição (ii). . . . .	27
2.8	(a) Um grafo fortemente cordal e (b),(c),(d),(e) um esquema de eliminação forte $(6, 3, 5, 2, 4, 1)$ para esse grafo. . . . .	28
2.9	Árvore de cliques de $G'$ . . . . .	31
2.10	Grafo $G'$ obtido a partir da instância $I = (X, C) = (\{x_1, x_2, \overline{x_1}, \overline{x_2}\}, \{(x_1 \vee \overline{x_2}), (x_2 \vee \overline{x_1})\})$ de MAX-2-SAT. . . . .	32
2.11	Esquema de eliminação forte para o grafo fortemente cordal $G'$ : (a) Eliminação dos conjuntos $D^{(i)}$ e $E^{(i)}$ ; (b) Eliminação dos conjuntos $U$ e $Z$ ; (c) Eliminação dos conjuntos $V, W$ e $T^{(i)}$ ; (d) Eliminação dos conjuntos $Q$ e $S$ ; (e) Eliminação dos conjuntos $C^{(i)}$ e posterior eliminação da clique $X \cup Y \cup R$ . . . . .	34
3.1	(a) Um grafo split e (b) Um grafo (2,4)-split. . . . .	35

3.2	Exemplo de corte máximo para o grafo completo $K_6$ .	36
3.3	Exemplo de corte máximo para um grafo split com $ N(S)  \leq \lfloor \frac{n}{2} \rfloor$ .	37
3.4	Exemplo de corte aproximado para um grafo split com $ N(S)  \geq \lfloor \frac{n}{2} \rfloor$ :	
	(a) $k S  \leq 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ e (b) $k S  > 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ .	38
3.5	Grafo split no qual o algoritmo aproximativo tem razão de aproximação $\frac{2}{3}$ .	39
3.6	(a) Um grafo (3,4)-split cheio simples e (b) um grafo (3,4)-split cheio múltiplo.	40
3.7	(a) Um grafo 1-split e (b) seu corte máximo.	41
3.8	Partição de um conjunto $X$ em duas partes $Y$ e $Z$ . Possíveis escolhas de subconjuntos de $X$ com cardinalidade 4.	41
4.1	Exemplo de um grafo sinalizado.	45
4.2	Exemplo de um grafo sinalizado balanceado com sua partição.	46
4.3	(a) Um grafo sinalizado não balanceado e (b) seu subgrafo balanceado máximo.	46
4.4	Partição de $V(K_5)$ com $\frac{m}{2} + \frac{n-1}{4} = \frac{10}{2} + \frac{5-1}{4} = 6$ arestas.	47
4.5	(a) Um ciclo positivo e (b) um ciclo negativo.	49
4.6	(a) Um grafo sinalizado balanceado e (b) sua partição balanceada.	49
4.7	Exemplo de um grafo bloco.	50
4.8	(a) Regras de redução <i>two-way</i> (b) Regras de redução <i>one-way</i> .	51
4.9	(a) Um bloco $C$ com $C_{\text{int}}$ e $C_{\text{ext}}$ , (b) blocos folha de um grafo e (c) blocos caminho e o vértice de caminho $v$ .	54
4.10	(a) Um bloco $C = \{x, y, u\}$ em $G - S$ nas condições da Regra 10. Se a aresta $\{x, y\}$ é uma ponte em $G - u$ , procedemos como em (b). Caso contrário, fazemos a transformação ilustrada em (c).	55
4.11	Grafo bloco $G - S$ sem arestas positivas obtido pelo algoritmo parametrizado FPT, mencionado na Proposição 4.3, para o problema do balanceamento devido à Crowston, Gutin, Jones e Muciaccia [15], obtido da entrada $(G, k)$ na qual $G$ é um grafo split sinalizado quando o algoritmo não responde expressamente SIM.	57
4.12	Único bloco caminho possível em $G - S$ , quando $K^2 = K_2$ e cada um dos vértices de $K^2$ tem pelo menos um vizinho em $I_1^2$ .	58
4.13	Vértices do conjunto $Q \subset I_1^2$ que não possui vizinhos em $S$ .	59
4.14	Redução para mostrar que MAX CUT é NP-difícil em grafos sem vértices universais.	66
4.15	Exemplo para $d = 3$ dos conjuntos definidos na demonstração da Afirmação 4.22.	69
5.1	(a) Um grafo aresta-colorido $G$ e (b) Um corte $\partial S$ de $G$ com $S = \{a, b\}$ e capacidade $C(\partial S) = 2$ .	72

5.2 (a) Um corte colorido mínimo com as partes circuladas e que apresenta muitas arestas(3) e poucas cores(1) e (b) um corte colorido máximo de um grafo com muitas cores(2) e poucas arestas(2). . . . .	73
5.3 (a) Grafo bipartido $G'$ obtido a partir da instância $((C, S), k)$ = $((\{1, 2, 3, 4, 5\}, \{S_1, S_2, S_3, S_4\}), 2)$ = $((\{1, 2, 3, 4, 5\}, \{\{1, 2\}, \{1, 4\}, \{1, 3, 5\}, \{2, 4, 5\}\}, 2)$ usado na redução de SET COVER e (b) o $(s, t)$ -grafo $G^c$ . Veja que existe uma cobertura de conjuntos $S_3, S_4$ de tamanho 2 para $C = \{1, 2, 3, 4, 5\}$ se, e somente se, existe um $(s, t)$ -corte de $(G^c, s, t, k' = 2)$ contendo as $k' = k = 2$ cores $S_3$ e $S_4$ . . . . .	76
5.4 (a) O grafo cúbico $G$ e (b) seu $(s, t)$ -grafo $G^c$ . . . . .	77
5.5 (a) Um grafo cúbico $G$ e (b) seu grafo associado $G'$ para o Lema 5.5. . . . .	78
5.6 (a) O grafo com grau máximo 3, $G = (V, E)$ no qual toda aresta incide a no máximo um vértice de grau 3, (b) seu $(s, t)$ -grafo intermediário $G'$ e (c) o grafo $G''$ no qual cada cor aparece no máximo 2 vezes e cada $(s, t)$ -caminho tem comprimento no máximo 3. . . . .	80
5.7 (a) O grafo $G^c$ modificado e (b) o grafo completo $G''$ no qual as arestas pontilhadas representam as arestas coloridas com a cor $p + 1$ . . . . .	82
5.8 (a) Os grafos $G_i$ e (b) o grafo $G$ construído no Lema 5.11. . . . .	84
5.9 Exemplo de um $(s, t)$ -grafo no qual cada cor aparece em no máximo dois $(s, t)$ -caminhos. . . . .	89
5.10 Exemplos de grafos tais que para todo par de vértices distintos $s, t \in V(G)$ , sempre existe um $(s, t)$ -corte de arestas não colorido de tamanho constante: os grafos com grau máximo limitado, $k$ -regulares e as árvores fazem parte dessa classe. . . . .	90
5.11 Um grafo planar com um vértice $v$ de grau menor ou igual a 5. . . . .	90
5.12 (a) Instância $G'$ de MAX-COLORED-CUT e (b) o grafo completo $G''$ . . . . .	91
5.13 Grafo $G'$ associado à instância $(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$ . . . . .	92
5.14 Multigrafo $G$ associado ao grafo $G'$ com arestas coloridas. . . . .	93
5.15 Corte de arestas do multigrafo $G$ associado à instância $(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$ com atribuição $x_1 = V$ , $x_2 = V$ e $x_3 = V$ . . . . .	94
5.16 Grafo simples $H$ associado ao multigrafo $G$ . . . . .	95
5.17 Corte de arestas associado ao grafo $H$ . . . . .	95
5.18 Grafo $G''$ . . . . .	96
5.19 Grafo $G^c$ associado à instância $I = (U, C)$ na qual $U = \{u_1, u_2, u_3\}$ e $C = \{(u_1 \vee u_2 \vee \overline{u_3}), (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3}), (\overline{u_1} \vee u_2 \vee \overline{u_3})\}$ . . . . .	97
5.20 Corte com 14 cores obtido de $G^c = (V, E, f)$ da Figura 5.19 para a atribuição satisfatível $\overline{u_1} = u_2 = u_3 = V$ . . . . .	99
5.21 Grafo $F$ obtido após a fusão dos vértices do multigrafo $G$ . . . . .	100

# **Lista de Tabelas**

5.1	Tabela apresentando um resumo dos resultados de complexidade para MIN-COLORED- $(s, t)$ -CUT. . . . .	91
6.1	C.C.=“complexidade clássica”; C.P.=“complexidade parametrizada”; O=“aberto”; P=“polinomial”. . . . .	107

# Capítulo 1

## Introdução

Um dos problemas básicos envolvendo grafos é o problema da determinação de um corte de arestas máximo (MAXCUT): dado um grafo simples  $G = (V, E)$  e uma função  $peso\ c : E \rightarrow \mathbb{N}$  que atribui a cada aresta de  $G$  um valor numérico chamado *peso da aresta*, determinar uma partição  $(S, V \setminus S)$  do conjunto de vértices  $V$ , de modo que a soma dos pesos das arestas com um dos extremos em  $S$  e o outro em  $V \setminus S$  seja a maior possível. Quando atribuímos um peso unitário a todas as arestas do grafo, o problema se reduz a determinar o corte de arestas de cardinalidade máxima. Esse problema é conhecido como *Problema do Corte Máximo Simples* ou SIMPLE MAXCUT.

Esses são problemas combinatórios clássicos da Teoria dos Grafos que têm sido amplamente estudados desde a década de 60. Ao final dessa década, mais precisamente em 1967, na busca por um meio de determinação do corte máximo simples, Erdős [26] mostrou que é sempre possível obter um corte de arestas com pelo menos metade das arestas do grafo. Com o desenvolvimento da Teoria da Computação, na década seguinte, em 1971, Cook introduziu o conceito de problemas de decisão *NP-completos* e no ano seguinte Karp provou que MAXCUT é NP-completo através de uma redução de PARTITION. Um pouco mais tarde, em 1976, Garey, Johnson e Stockmeyer mostraram que SIMPLE MAXCUT é também NP-completo. A partir daí foram realizados esforços no sentido de encontrar algoritmos aproximativos polinomiais que resolvessem o problema do Corte Máximo Simples com a maior eficiência possível e, ao mesmo tempo, a complexidade desse problema passou a ser estudada em classes de grafos particulares, com o objetivo de determinar se SIMPLE MAXCUT tem solução polinomial em alguma classe de grafos específica. Por exemplo, em 1975, Hadlock [45] mostrou que SIMPLE MAXCUT é polinomial em grafos planares. Esse mesmo resultado foi obtido algum tempo antes por Orlova e Dorfman [52], em 1972. Em 1995, Goemans e Williamson [42] encontraram um algoritmo aproximativo polinomial para SIMPLE MAXCUT cuja razão de aproximação é 0,878, e complexidade de tempo  $O(n^3)$ . O algoritmo aproximativo de Goemans e Williamson [42] usa pro-

gramação semidefinida [67] e portanto, não é “puramente” combinatório. Assim, a motivação na literatura para algoritmos aproximativos polinomiais combinatórios para SIMPLE MAXCUT com razão de aproximação inferior e tempo de execução mais eficiente para grafos em geral ou para classes particulares de grafos ainda se encontra presente.

O problema do Corte Máximo pode ser aplicado para resolver o problema da análise de *Clusters*, no qual um conjunto de dados representados por pontos é partitionado em grupos (*clusters*) de pontos de modo que em cada *cluster* os pontos são “similares” e dois pontos de um mesmo *cluster* estão mais próximos do que dois pontos em *clusters* distintos. Esse problema pode ser formulado matematicamente, criando-se um grafo no qual cada vértice representa um ponto e cada par de vértices é conectados por uma aresta cujo peso é determinado pela proximidade dos vértices ligados. Considerando-se dados numéricos, por exemplo, a proximidade entre os pontos pode ser definida pela distância Euclidiana. Determinar um corte máximo no grafo que modela o problema significa encontrar dois *clusters* cujos pontos dentro de cada um deles são os mais “similares” (próximos) possíveis. Barahona, Grötschel, Jünger e Reinelt [2] descreveram como o problema do Corte Máximo surge no *design* de circuitos VLSI e na análise dos modelos *spin-glass*.

## 1.1 Organização do texto

No Capítulo 1 apresentamos, além da introdução, as definições básicas de Teoria dos Grafos necessárias ao entendimento geral do texto, bem como as notações utilizadas ao longo de toda a tese. Apresentamos também as preliminares e o histórico dos resultados básicos conhecidos para o Problema do Corte Máximo Simples (SIMPLE MAXCUT). No Capítulo 2, provamos que SIMPLE MAX CUT é NP-completo para grafos fortemente cordais, partindo da construção realizada por Bodlaender e Jansen [6] e mostrando que o grafo resultante é fortemente cordal através da exibição de um esquema de eliminação simples para o grafo construído. Neste mesmo artigo de Bodlaender e Jansen [6], foi provado que SIMPLE MAX CUT é NP-completo para grafos split e, com base nesse resultado, foi feito o estudo do Capítulo 3. Como SIMPLE MAX CUT em grafos split é NP-completo, exibimos um algoritmo  $\frac{2}{3}$ -aproximativo polinomial para a determinação do corte máximo em grafos split. Em seguida, ainda neste capítulo, definimos a subclasse dos grafos  $(k, n)$ -split cheios e mostramos que para essa subclasse de grafos, SIMPLE MAXCUT está em P. No Capítulo 4 exploramos uma nova abordagem para a solução de SIMPLE MAX CUT através do estudo do problema SIGNED MAX CUT no qual é dado um grafo sinalizado e procura-se determinar o subgrafo sinalizado balanceado desse grafo que possua o maior número de arestas. No início do Capítulo 4 mostramos que SIMPLE MAX CUT é um caso

particular desse problema e passamos a estudar SIGNED MAX CUT do ponto de vista da Complexidade Parametrizada, considerando as classes de grafos split e grafos  $(r, \ell)$ . Mostramos que SIGNED MAX CUT tem núcleo quadrático nessas duas classes e encontramos uma subclasse de grafos split – os grafos  $d^*$ -split – na qual SIGNED MAX CUT tem núcleo linear. Buscando novas abordagens para o problema de cortes, no Capítulo 5 consideramos os problemas cuja entrada é um grafo aresta-colorido  $G^c = (V, E, f)$ , no qual  $f : E \rightarrow \{1, 2, 3, \dots, c\}$  é uma coloração de arestas dada, e procuramos cortes de arestas específicos que apresentem o menor e o maior número de cores possíveis, respectivamente. O primeiro deles é o problema do corte colorido mínimo (MIN-COLORED-CUT) em que buscamos um corte de arestas que apresenta o menor número de cores; e o segundo é o problema do corte colorido máximo (MAX-COLORED-CUT) no qual se busca um corte de arestas com o maior número de cores presentes.

## 1.2 Definições Básicas e Notações

Nesta seção fazemos as definições básicas da Teoria dos Grafos utilizadas ao longo de toda a tese e que podem ser encontradas no livro de Bondy e Murty [8]. Além disso, apresentamos três subsessões referentes aos temas de Complexidade Clássica, Complexidade Parametrizada e Algoritmos Aproximativos.

Um *grafo*  $G = (V, E, \psi)$  é uma estrutura matemática formada por um conjunto finito não vazio  $V$  cujos elementos são chamados *vértices*, um conjunto  $E$  cujos elementos são chamados *arestas* e uma *função de incidência*  $\psi : E \rightarrow V \times V$  que associa a cada aresta  $e \in E$  um par não ordenado  $\{u, v\}$  de vértices de  $V$ . Se  $\psi(e) = \{u, v\}$ , dizemos que os vértices  $u$  e  $v$  são os *extremos* da aresta  $e$ , ou que a aresta  $e$  *liga* os vértices  $u$  e  $v$ , ou ainda que os vértices  $u$  e  $v$  são *adjacentes*, ou que a aresta  $e$  é *incidente* aos vértices  $u$  e  $v$ . Uma aresta que liga um vértice  $v$  a si mesmo é chamada *laço*, isto é,  $e \in E$  é um laço se  $\psi(e) = \{v, v\}$ . Analogamente, duas ou mais arestas que ligam o mesmo par de vértices são ditas *paralelas* ou *múltiplas*, ou seja,  $e_1$  e  $e_2$  são arestas paralelas se  $\psi(e_1) = \psi(e_2)$ .

Um grafo que não admite laços e nem arestas paralelas é dito um *grafo simples*. Neste caso representamos apenas  $G = (V, E)$  e como não há perigo de confusão, associamos diretamente  $e = \{u, v\}$  no lugar de  $\psi(e) = \{u, v\}$ . Um grafo que possui laços ou arestas paralelas é chamado *multigrafo*.

Dado um grafo  $G = (V, E)$  e  $v \in V(G)$ , a *vizinhança aberta* de  $v$  é o conjunto formado por todos os vértices de  $G$  adjacentes a  $v$ , que representaremos por  $N(v)$ , ou seja,  $N(v) = \{u \in V(G) | \{u, v\} \in E(G)\}$ . Analogamente, a *vizinhança fechada* de  $v$ , que representaremos por  $N[v]$  é definida como  $N[v] = N(v) \cup \{v\}$ .

Um vértice  $v \in V(G)$  é dito *universal* se é adjacente a todos os outros vértices

de  $G$ , isto é, se  $N[v] = V(G)$ . Analogamente, um vértice  $v \in V(G)$  é dito *isolado* quando não é adjacente a nenhum vértice de  $G$ , ou seja,  $N[v] = \{v\}$ .

O *grau* de um vértice  $v \in V(G)$  é dado pelo número de arestas incidentes a  $v$  em  $G$ . No caso de multigrafos com laços, cada laço incidente a  $v$  contribui com duas unidades para o grau de  $v$  e no caso de grafos simples, o grau de  $v$  é dado pela cardinalidade do conjunto  $N(v)$ . Representamos o grau do vértice  $v$  por  $d(v)$ . Temos então  $d(v) = |N(v)|$  em grafos simples.

Um grafo simples  $G$  é dito *k-regular* se todos os seus vértices possuem o mesmo grau  $k$ . Os grafos 3-regulares são chamados grafos *cúbicos*.

Um grafo  $G$  é dito *completo* se todos os seus vértices são universais. Representamos o grafo completo com  $n$  vértices por  $K_n$ .

Um grafo  $G$  é dito *nulo* se todos os seus vértices são isolados.

Um grafo  $H = (V', E')$  é dito um *subgrafo* de  $G = (V, E)$  se  $V' \subseteq V$ ,  $E' \subseteq E$ .

Dados um grafo  $G = (V, E)$  e  $S \subseteq V$ , não vazio, o *subgrafo induzido* por  $S$ , que representaremos por  $G[S]$  é o subgrafo de  $G$  cujo conjunto de vértices é dado por  $S$  e cujo conjunto de arestas é composto por todas as arestas de  $G$  com ambos os extremos em  $S$ .

Dado um grafo  $G = (V, E)$ , dizemos que um conjunto  $K \subseteq V(G)$  é uma *clique* de  $G$  se  $K$  induz um subgrafo de  $G$  que é completo, ou seja, se  $G[K]$  é completo. Do mesmo modo,  $I \subseteq V(G)$  é um *conjunto independente* de  $G$  se  $G[I]$  é nulo.

Um grafo  $G = (V, E)$  é uma *estrela* se  $V = \{u, v_1, v_2, \dots, v_n\}$ , sendo  $u$  um vértice universal e  $\{v_1, v_2, \dots, v_n\}$  um conjunto independente.

Um grafo  $G = (V, E)$  é dito *bipartido* se o conjunto de vértices  $V$  puder ser particionado em dois subconjuntos  $V_1$  e  $V_2$  não vazios tais que toda aresta de  $G$  tem um extremo em  $V_1$  e o outro em  $V_2$ . Um grafo  $G$  é *bipartido completo* se  $G$  é bipartido e todo par de vértices  $(v_1, v_2) \in V_1 \times V_2$  é ligado por uma aresta de  $G$ . Se  $|V_1| = n_1$  e  $|V_2| = n_2$ , representamos o grafo bipartido completo correspondente por  $K_{n_1, n_2}$ . Os grafos bipartidos completos são também chamados de *bicliques*.

Um *caminho* em um grafo  $G$  é uma sequência  $P = (v_1, v_2, \dots, v_p)$  de vértices dois a dois distintos de  $G$  tais que  $v_i$  é adjacente a  $v_{i+1}$  para todo  $i \in \{1, \dots, p-1\}$ , isto é,  $\{v_i, v_{i+1}\} \in E(G)$  para todo  $i \in \{1, \dots, p-1\}$ . Os vértices  $v_1$  e  $v_p$  são chamados *extremos* do caminho  $P$ . Dizemos também que o caminho  $P$  liga  $v_1$  a  $v_p$ . O *comprimento* do caminho  $P$  é dado pelo número de arestas de  $P$ .

Um *ciclo* em um grafo  $G$  é um caminho fechado, isto é, um caminho no qual  $v_p = v_1$ . De forma genérica, representamos um ciclo com  $n$  vértices por  $C_n = (v_1, v_2, \dots, v_n, v_1)$ . Um *ciclo par* é um ciclo que possui um número par de arestas. Caso contrário temos um *ciclo ímpar*. Os ciclos são exemplos de grafos 2-regulares. Um ciclo que contém todos os vértices do grafo  $G$  é dito um *ciclo hamiltoniano* de  $G$ .

Um conjunto  $S$  é dito *maximal* com relação a uma propriedade  $\Pi$  se  $S$  satisfaz  $\Pi$  e não existe nenhum conjunto  $S'$  que contém propriamente  $S$  e satisfaz  $\Pi$ . Analogamente,  $S$  é dito *minimal* com relação a uma propriedade  $\Pi$  se  $S$  satisfaz  $\Pi$  e não existe nenhum conjunto  $S'$  propriamente contido em  $S$  que satisfaz  $\Pi$ .

Um grafo  $G$  é dito *conexo* se para todo par  $u, v$  de vértices distintos de  $G$  existe pelo menos um caminho ligando  $u$  a  $v$ . Uma *componente conexa* de  $G$  é um subgrafo conexo maximal de  $G$ . Um grafo  $G$  é conexo se, e somente se, possui uma única componente conexa que é ele mesmo. Caso o grafo  $G$  não seja conexo, dizemos que  $G$  é *desconexo*, ou seja,  $G$  possui pelo menos duas componentes conexas. Uma aresta  $e \in E$  de um grafo conexo  $G$  é uma *ponte* se o grafo  $G - e$  é desconexo, ou seja, se a remoção de  $e$  dá origem a um grafo desconexo.

Dizemos que uma classe de grafos  $\mathcal{C}$  é *hereditária* se todo subgrafo induzido de um grafo  $G$  em  $\mathcal{C}$  também pertence a  $\mathcal{C}$ .

Uma *coloração de vértices* de um grafo  $G = (V, E)$  é uma função  $c : V \rightarrow \mathbb{N}$  que associa a cada aresta  $e \in E$  uma *cor*  $c(e) \in \mathbb{N}$ . Uma coloração de vértices de um grafo  $G$  é dita *própria* se vértices adjacentes em  $G$  possuem sempre cores distintas, ou seja, se  $\{u, v\} \in E$ , então  $c(u) \neq c(v)$ . Um grafo  $G$  é dito *k-cromático* se  $G$  possui uma coloração própria com  $k$  cores.

Um *hipergrafo*  $H = (V, E)$  consiste de um conjunto finito  $V$  de vértices e um conjunto  $E$  de hiperarestas no qual cada *hiperaresta* é um subconjunto de  $V$ . Um grafo simples  $G = (V, E)$  é um hipergrafo no qual cada hiperaresta é um subconjunto de  $V$  com cardinalidade fixa 2.

### 1.2.1 Complexidade Clássica

As definições apresentadas nessa seção foram baseadas nos textos de [60] e [37].

Um *problema computacional*  $\Pi$  é composto por uma questão a ser resolvida, envolvendo uma ou mais variáveis de entrada. Uma *instância* de  $\Pi$  é um conjunto particular de valores assumidos pelas variáveis de entrada do problema  $\Pi$ .

Um problema  $\Pi$  é dito *problema de decisão*, se  $\Pi$  admite como possíveis respostas apenas SIM ou NÃO. Desse modo, a formulação de um problema de decisão é composta por uma descrição geral de suas variáveis de entrada, seguida por uma questão cujas possíveis respostas são SIM ou NÃO, proposta em função dessas variáveis. Considerando o problema de decisão  $\Pi$  com um conjunto de instâncias  $I_\Pi$ , podemos observar que existe uma partição de  $I_\Pi$  em dois subconjuntos  $S_\Pi$  e  $N_\Pi$  que são, respectivamente, os conjuntos de instâncias que dão a resposta SIM e NÃO ao problema  $\Pi$ , ou seja,  $N_\Pi = I_\Pi \setminus S_\Pi$ .

Um *algoritmo*  $A$  que soluciona o problema  $\Pi$  é uma sequência finita de instruções que aplicada a qualquer instância  $x \in I_\Pi$  produz uma solução  $y$  de  $\Pi$  associada à

instância  $x$ , ou seja,  $y = A(x)$ . Diremos que  $A$  é um *algoritmo determinístico* para  $\Pi$  se dada qualquer instância  $x \in I_\Pi$  tem-se que  $A(x)$  realiza sempre a mesma sequência de passos e produz sempre a mesma solução para a instância  $x$ . Caso contrário, isto é, quando sucessivas aplicações do algoritmo  $A$  a uma mesma instância  $x$  produzem soluções distintas, diremos que  $A$  é um *algoritmo randomizado*.

Sejam  $\Pi$  um problema computacional,  $A$  um algoritmo que resolve  $\Pi$  e  $I_\Pi(n)$  o conjunto de todas as instâncias de  $\Pi$  com tamanho  $n$ . Definimos a *complexidade de tempo* do algoritmo  $A$  como a função que associa a cada inteiro positivo  $n$ , o tempo máximo que  $A$  demanda para fornecer uma solução de  $\Pi$ , qualquer que seja a instância de entrada  $x \in I_\Pi(n)$ .

Dizemos que uma função  $f(n)$  é *da ordem de*  $g(n)$  e escrevemos  $f(n) = O(g(n))$  quando existem um natural  $n_0$  e uma constante  $c > 0$  tais que  $0 \leq f(n) \leq c \cdot g(n)$  para todo  $n \geq n_0$ .

Dados  $\Pi$  um problema computacional,  $A$  um algoritmo que resolve  $\Pi$  e  $I_\Pi(n)$  o conjunto de todas as instâncias de  $\Pi$  com tamanho  $n$ , dizemos que o algoritmo  $A$  possui *complexidade de tempo polinomial*, ou simplesmente que  $A$  é *polinomial*, se existir um polinômio  $p(n)$  tal que a função de complexidade de tempo do algoritmo  $A$  é  $O(p(n))$  para toda instância de entrada em  $I_\Pi(n)$ .

Um problema  $\Pi$  pertence à *classe de problemas solucionáveis em tempo polinomial* denotada por  $P$  se existe um algoritmo determinístico com complexidade de tempo polinomial que soluciona  $\Pi$ . Dizemos também que  $\Pi$  *está em*  $P$  e representamos por  $\Pi \in P$ .

Um problema de decisão  $\Pi$  pertence à *classe dos problemas NP* se existe um algoritmo determinístico de tempo polinomial que verifica a validade de um certificado polinomial para o SIM. Analogamente, se existir um algoritmo determinístico de tempo polinomial que verifique a validade de um certificado polinomial para o NÃO, diremos que  $\Pi$  pertence à *classe dos problemas coNP*.

Uma *redução polinomial* de um problema de decisão  $\Pi$  para um problema de decisão  $\Pi'$  é uma função  $f : I_\Pi \rightarrow I_{\Pi'}$  que satisfaz as seguintes condições:

- existe um algoritmo determinístico polinomial  $A$  que computa  $f$ ;
- $x \in S_\Pi$  se, e somente se,  $f(x) \in S_{\Pi'}$ .

Neste caso dizemos também que  $\Pi$  se *reduz polinomialmente* a  $\Pi'$ , e denotamos essa relação por  $\Pi \propto \Pi'$ . Uma consequência imediata da existência de uma redução polinomial de  $\Pi$  para  $\Pi'$  é que se  $\Pi' \in P$ , então  $\Pi \in P$ .

Um problema  $\Pi$  é *NP-difícil* se existe um problema  $\Pi' \in NP$  tal que  $\Pi' \propto \Pi$ . Caso  $\Pi$  seja NP-difícil e ainda  $\Pi \in NP$ , dizemos que  $\Pi$  é *NP-completo*.

### 1.2.2 Algoritmos Aproximativos

Nesta seção apresentamos alguns conceitos relacionados a algoritmos aproximativos polinomiais [68].

**Definição.** [68] (*Problemas de Otimização*) Consideremos um problema combinatório  $A$  com uma instância  $x_A$  e um conjunto  $S_A(x_A)$  de soluções viáveis do problema para essa instância. O *custo*  $c_A(x_A, s)$  de uma solução viável  $s$  associada à instância  $x_A$  é um número inteiro não negativo. Por exemplo, no caso de SIMPLE MAX CUT, o custo de uma solução é dado pelo número de arestas presentes no corte obtido. Uma *solução ótima*  $OPT_A(x_A)$  de um problema de maximização (minimização)  $A$  com relação a uma instância  $x_A$  é uma solução viável de  $A$  cujo custo é o máximo (mínimo) possível no conjunto de soluções viáveis  $S_A(x_A)$ .

**Definição.** [20] (*S-redução*) Sejam  $A$  e  $B$  dois problemas de otimização. Consideremos  $x_A$  e  $x_B$  instâncias de  $A$  e  $B$  com conjuntos de soluções viáveis  $S_A(x_A)$  e  $S_B(x_B)$  com custos  $c_A(x_A, s)$  e  $c_B(x_B, s')$ , sendo  $s \in S_A(x_A)$  e  $s' \in S_B(x_B)$ , respectivamente. Sejam ainda  $OPT_A(x_A)$  e  $OPT_B(x_B)$  as soluções ótimas dos problemas  $A$  e  $B$ . Uma *S-redução* é um par  $f$  e  $g$  de algoritmos de tempo polinomial no tamanho de  $x_A$ , tais que:

1.  $f$  leva instâncias  $x_A$  de  $A$  em instâncias  $x_B = f(x_A)$  de  $B$ , e  $g$  leva instâncias  $x_B$  de  $B$  em instâncias  $x_A = g(x_B)$  de  $A$ ;
2. Se  $y'$  é uma solução viável de  $x_B$ , então  $g(y')$  é uma solução viável de  $x_A$ ;
3.  $c_B(f(x_A), OPT_B(f(x_A))) = c_A(x_A, OPT_A(x_A))$ ;
4.  $c_A(x_A, g(y')) = c_B(x_B, y')$ .

Se um problema de otimização  $A$  se S-reduz a outro problema de otimização  $B$  e sabemos resolver  $B$  por um algoritmo exato de tempo polinomial, então é possível transformar uma instância de  $A$  em uma instância de  $B$  através da S-redução, resolver o problema para a instância de  $B$  encontrada utilizando o algoritmo polinomial conhecido e transformar a solução obtida para  $B$  numa solução de  $A$  utilizando novamente a S-redução. Caso  $B$  seja um problema NP-completo, a existência da S-redução de  $A$  em  $B$  nos garante que  $A$  também é NP-completo. Esses resultados podem ser resumidos na Proposição 1.1. Ainda mais, no caso de problemas de decisão NP-completos ou problemas de otimização NP-difíceis, não se conhece algoritmo polinomial que dê uma solução exata do problema considerado e lançamos mão de algoritmos que dão uma solução aproximada do problema.

Um exemplo de S-redução é dado entre os problemas  $A = \text{CLIQUE MÁXIMA}$  e  $B = \text{CONJUNTO INDEPENDENTE MÁXIMO}$ . Se  $x_A = G = (V, E)$  é uma instância de  $A$ , e  $x_B = \overline{G} = (V, \overline{E})$  é uma instância de  $B$ , então o algoritmo polinomial  $f : \mathcal{G} \rightarrow \mathcal{G}$ , em que  $\mathcal{G}$  é a classe de todos os grafos, definido por  $f(G) = \overline{G}$ , leva instâncias de  $A$  em  $B$ , de modo que se  $S \subset V(\overline{G})$  é uma solução de CONJUNTO INDEPENDENTE MÁXIMO em  $\overline{G}$ , então  $g(S) = S$  é uma solução de CLIQUE MÁXIMA em  $G$ . A Figura 1.1 ilustra essa S-redução.

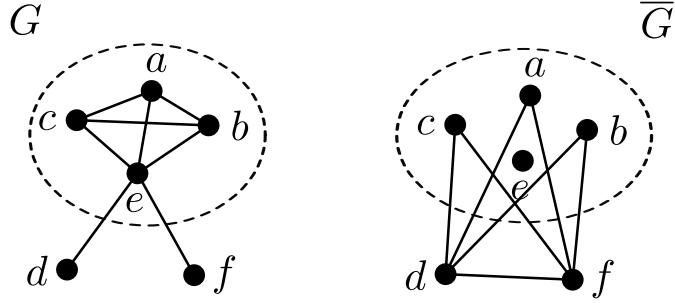


Figura 1.1: Um grafo  $G$  com sua clique máxima e o grafo complementar  $\overline{G}$  com seu conjunto independente máximo.

Um *algoritmo de aproximação* é um algoritmo polinomial usado para encontrar uma solução viável de um problema de otimização que está “próxima” da solução ótima do problema de otimização considerado. Um *Esquema de Aproximação em Tempo Polinomial (PTAS)* é um algoritmo que toma como entrada um problema de otimização  $\Pi$  e um parâmetro  $\varepsilon > 0$  chamado *fator de aproximação* e, em tempo polinomial, produz uma solução  $S$  de  $\Pi$  tal que  $|S - OPT(\Pi)| \leq \varepsilon \cdot |OPT(\Pi)|$ , sendo  $OPT(\Pi)$  a solução ótima do problema  $\Pi$ . Alguns problemas que não têm um *PTAS* podem admitir um algoritmo randomizado com propriedades semelhantes. Um *Esquema de Aproximação em Tempo Polinomial Randomizado (PRAS)* é um algoritmo que toma como entrada um problema de otimização  $\Pi$  e um parâmetro  $\varepsilon > 0$  e, em tempo polinomial, produz uma solução  $S$  de  $\Pi$  tal que  $Pr [|S - OPT(\Pi)| \leq \varepsilon \cdot |OPT(\Pi)|] \geq \frac{1}{2}$ .

Uma *classe de complexidade de aproximação* é uma classe de problemas de otimização que possuem as mesmas propriedades approximativas [13].

**Proposição 1.1.** [12]. *Sejam  $A$  e  $B$  dois problemas de otimização, se  $A$  se S-reduz a  $B$  e  $B$  pertence a uma classe  $\mathcal{C}$  de complexidade de aproximação, então  $A \in \mathcal{C}$ .*

### 1.2.3 Complexidade Parametrizada

A seguir apresentamos alguns conceitos da Complexidade Parametrizada que serão utilizados nos Capítulos 4 e 5. Um problema de otimização  $\Pi$  é dito *parametrizado* quando uma variável  $k$  presente na instância do problema é escolhida como *parâmetro*.

**Definição.** [21] (*Classe FPT*) Dado um problema  $A$ , dizemos que  $A$  é *tratável por parâmetro fixo*, i.e., *pertence à classe FPT* com relação a um parâmetro  $k$  (escolhido em  $A$ ) ou é *computável em tempo FPT* com relação a  $k$ , se existe um algoritmo que resolve  $A$  em tempo  $f(k) \cdot n^{O(1)}$  para alguma função computável  $f$ .

**Definição.** [21] (*FPT-redução*) Dados dois problemas parametrizados  $(A, k)$  e  $(B, k')$ , uma *FPT-redução* é uma transformação que cumpre as seguintes condições:

1. leva uma instância  $(x_1, k)$  de  $A$  em uma instância  $(x_2, k')$  de  $B$ ;
2.  $(x_1, k) \in A$  se e somente se  $(x_2, k') \in B$ ;
3.  $(x_2, k')$  é computável em tempo FPT de  $k$
4.  $k' \leq g(k)$  para alguma função computável  $g$ .

Além da classe e da redução FPT, Downey e Fellows [20] definiram classes de problemas parametrizados de acordo com seu nível de intratabilidade parametrizada. Essas classes são organizadas em uma *W-hierarquia* ( $FPT \subseteq W[1] \subseteq W[2] \subseteq \dots \subseteq W[P] \subseteq \dots$ ), e são baseadas na complexidade dos circuitos necessários para se verificar a validade de uma solução, ou, alternativamente, na profundidade natural dos circuitos lógicos para o problema. É conjecturado que cada uma dessas inclusões de classes são próprias, e se  $P = NP$  então  $FPT = W[P]$ , veja [60].

Seja  $C$  um *circuito booleano de decisão* (para maiores definições, consulte [21]) com variáveis de entrada  $x_1, x_2, \dots, x_n$ . Uma porta lógica de  $C$  é dita *larga* se o seu número de entradas excede algum limite constante, usualmente 2. O *entrelaçamento* de  $C$  é o número máximo de portas lógicas largas em qualquer caminho a partir das variáveis de entrada até a linha de saída de  $C$ . A *profundidade* de  $C$  é o comprimento do maior caminho de uma variável de entrada até uma linha de saída em  $C$ . O *peso* de uma atribuição de valores-verdade para as variáveis de  $C$  é o número de variáveis definido com valor *verdadeiro* nessa atribuição.

Abaixo definimos o problema de SATISFATIBILIDADE PONDERADA EM CIRCUITOS, conhecido abreviadamente por  $WCS[t, h]$ .

SATISFATIBILIDADE PONDERADA EM CIRCUITOS ( $WCS[t, h]$ )

INSTÂNCIA: Um circuito de decisão  $C$  com entrelaçamento  $t$  e profundidade  $h$  e um inteiro não negativo  $p$ .

PARÂMETRO: O inteiro  $p$ .

QUESTÃO: Existe uma atribuição de valores-verdade para as variáveis de  $C$  que satisfaça o circuito  $C$  e tenha peso  $p$ ?

Para  $t \geq 1$ , dizemos que um problema parametrizado  $\Pi$  pertence à classe  $W[t]$  se, e somente se, existe uma redução parametrizada de  $\Pi$  para  $WCS[t, h]$ , para alguma constante  $h \geq 1$ .

Um problema parametrizado  $\Pi$  é dito  $W[t]$ -*difícil* se existe uma FPT-redução de  $\Pi$  para algum problema em  $W[t]$ . Se além dessa condição tivermos  $\Pi \in W[t]$ , diremos que  $\Pi$  é  $W[t]$ -*completo*.

A seguir apresentamos a Proposição 1.2 devida a Downey e Fellows.

**Proposição 1.2.** [21] *Sejam  $(A, k)$  e  $(B, k')$  dois problemas parametrizados, se  $A$  se FPT-reduz a  $B$  e  $B$  pertence a uma classe  $\mathcal{C}$  da  $W$ -hierarquia, então  $A \in \mathcal{C}$ .*

É conhecido na literatura especializada que todo problema em FPT admite um núcleo (*kernel*), isto é, pode-se aplicar um conjunto de regras de redução que transformam a instância original em uma instância “equivalente” com tamanho dependente somente de  $k$ . Em teoria da complexidade parametrizada, frequentemente é possível reduzir em tempo polinomial a instância original do problema em uma instância cujo tamanho resultante seja não maior que o tamanho da instância original. Essa instância resultante é conhecida como núcleo. Além disso, o tamanho do núcleo é sempre limitado por uma função do parâmetro. Sendo assim, cabe perguntar se tais problemas em FPT admitem um núcleo de tamanho limitado por uma função polinomial com relação ao parâmetro escolhido. Todavia núcleos polinomiais nem sempre ocorrem como podemos observar no artigo de Bodlaender, Downey, Fellows e Hermelin [5]. E às vezes existem teoremas relacionando a sua possível existência.

### 1.3 Partições de Vértices e Cortes de Arestas

Dado um grafo  $G = (V, E)$ , uma *partição*  $(A, B)$  de  $V$  é uma decomposição de  $V$  como união disjunta de dois subconjuntos não vazios  $A$  e  $B$ , ou seja,  $V = A \cup B$  com  $A \cap B = \emptyset$ , sendo  $A \neq \emptyset$  e  $B \neq \emptyset$ . Se  $V' \subset V$  é não vazio, diremos que a partição  $(A, B)$  de  $V$  *induz* a partição  $(A', B')$  se  $A' \subset A$ ,  $B' \subset B$  e  $V' = A' \cup B'$ . Nesse caso, diremos que  $(A', B')$  é uma partição de  $V'$  *induzida* pela partição  $(A, B)$  de  $V$ .

Dados um grafo  $G = (V, E)$  e  $A, B \subset V$  não vazios, o conjunto formado pelas arestas que possuem um dos extremos em  $A$  e o outro em  $B$  será representado por  $[A, B]$ . O subconjunto de  $E$  formado pelas arestas que possuem apenas um dos extremos em  $S \subset V$ , o qual representaremos por  $[S, V \setminus S] = \partial S$  é chamado um *corte de arestas* de  $G$ . As arestas de  $\partial S$  são ditas *arestas de corte*. Observe que a remoção de todas as arestas de  $\partial S$  de  $G$  produz um subgrafo de  $G$  desconexo.

De modo genérico, num grafo  $G = (V, E)$  com  $|V| = n$ , podemos rotular os vértices com os inteiros de 1 a  $n$ , isto é,  $V = \{1, 2, \dots, n\}$  e assim representar as

arestas de  $G$  como pares não ordenados de vértices  $\{i, j\}$  com  $i < j$ . O peso de uma aresta  $e = \{i, j\}$  será representado por  $c_e$  ou  $c_{ij}$ . Se o peso das arestas de  $G$  for sempre positivo, considerando que todo grafo  $G$  com  $n$  vértices é subgrafo do grafo completo  $K_n$ , podemos considerar sempre o grafo completo  $K_n$  e atribuir o valor  $c_{ij} = 0$  caso os vértices  $i$  e  $j$  de  $K_n$  não sejam adjacentes em  $G$ .

Com base na notação anterior, podemos representar o corte de arestas  $\partial S$  como

$$\partial S = \{\{i, j\} \in E \mid i \in S, j \notin S\}.$$

O *peso* do corte  $\partial S$ , que representaremos por  $c(\partial S)$ , é a soma dos pesos das arestas presentes no corte, isto é,

$$c(\partial S) = \sum_{e \in \partial S} c_e = \sum_{i \in S, j \notin S} c_{ij}.$$

Dados um grafo  $G = (V, E)$  e uma função peso  $c : E \rightarrow \mathbb{N}$ , um *corte máximo* de  $(G, c)$  é um conjunto  $\partial S \subseteq E$  que cumpre a condição  $c(\partial S) \geq c(\partial S')$  para todo  $S' \subset V$ . O peso do corte máximo de  $(G, c)$  será representado por  $mc(G, c)$ :

$$mc(G, c) = \max_{S \subset V} c(\partial S).$$

No caso do corte máximo simples, isto é, quando  $c_e = 1$  para toda aresta  $e \in E(G)$ , escreveremos simplesmente  $mc(G)$  para representar a cardinalidade do corte máximo associado ao grafo  $G$ . A Figura 1.2 ilustra um grafo  $G$  com um corte de arestas de cardinalidade 9.

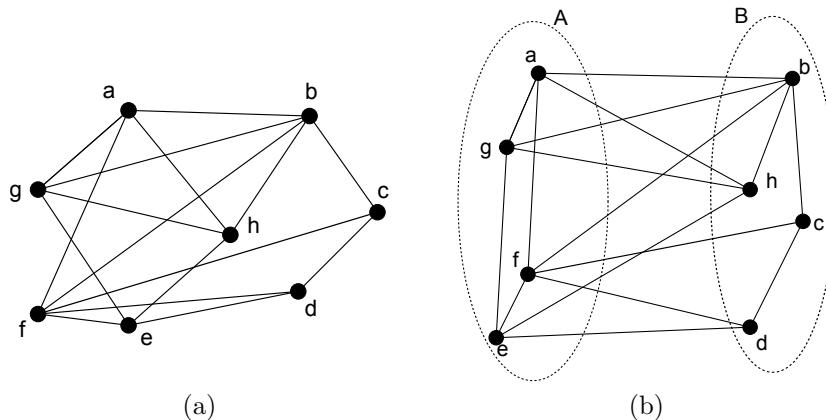


Figura 1.2: (a) Um grafo  $G$  e (b) um corte de arestas de  $G$  de custo 9.

Note que SIMPLE MAXCUT é trivial em grafos bipartidos. De fato, se  $G = (V, E)$  é um grafo *bipartido*, por definição  $V = X \cup Y$  com  $X$  e  $Y$  não vazios disjuntos e  $[X, X] = [Y, Y] = \emptyset$ , ou seja, temos uma partição  $(X, Y)$  de  $V$  e não existem arestas de  $G$  ligando vértices em uma mesma parte dessa partição. Portanto todas as arestas de  $G$  pertencem ao corte  $[X, Y]$  e trivialmente este é um corte máximo.

Determinar o corte máximo simples de um grafo  $G$  é equivalente a determinar o subgrafo bipartido de  $G$  com o maior tamanho (número de arestas) possível. Por sua vez, isso é equivalente a encontrar o subgrafo 2-cromático máximo de  $G$ .

## 1.4 Limite inferior para o corte máximo simples

Em 1967, Erdős [26] mostrou que para todo grafo  $G$  é sempre possível determinar um corte com pelo menos metade das arestas, usando o método probabilístico. Ele considerou aleatoriamente um subconjunto  $S \subset V$  e uma aresta  $\{u, v\}$ . A probabilidade de  $\{u, v\}$  pertencer a  $\partial S$  é  $\frac{1}{2}$ , (veja a Figura 1.3) e a linearidade do valor esperado nos diz que existe um corte com metade das arestas de  $G$ . De fato, considere uma enumeração de  $E(G) = \{e_1, e_2, \dots, e_m\}$  e para cada  $1 \leq i \leq m$  defina  $X_i = 1$  se a aresta  $e_i$  pertence ao corte  $\partial S$  e  $X_i = 0$  em caso contrário. Temos  $E[X_i] = \frac{1}{2}$ . Seja  $c(S)$  a variável aleatória que representa o tamanho do corte  $\partial S$ . Então  $E[c(S)] = E[\sum_{i=1}^m X_i] = m \cdot \frac{1}{2} = \frac{m}{2}$ . Como o valor esperado da variável aleatória  $c(S)$  é  $\frac{m}{2}$ , pelo método probabilístico existe  $S \subset V(G)$  tal que  $|\partial S| \geq \frac{m}{2}$ .

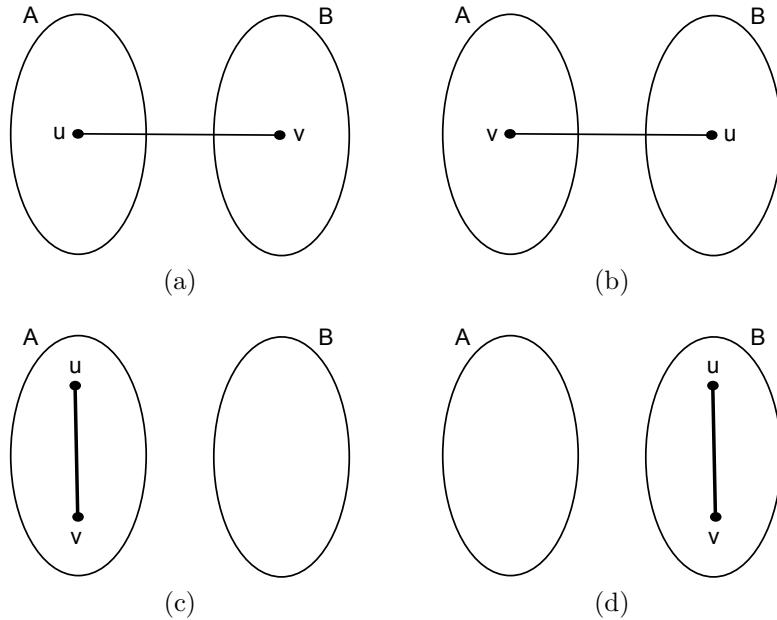


Figura 1.3: Quatro casos possíveis para a disposição dos vértices  $u$  e  $v$ . Somente os casos (a) e (b) são favoráveis

A partir desse resultado podemos construir um algoritmo randomizado Las Vegas que obtém um corte  $[A, V \setminus A]$  com tamanho esperado  $\frac{m}{2}$ . Considere uma ordenação  $(v_1, v_2, \dots, v_n)$  dos vértices de  $V$  e dois conjuntos  $A$  e  $B$  inicialmente vazios. Para cada vértice  $v_1, v_2, v_3, \dots, v_n$  executamos o procedimento randômico de posicionar  $v_i$  em  $A$  com probabilidade  $\frac{1}{2}$ .

O único problema é que o método das esperanças não nos fornece um limite

inferior para um corte com pelo menos  $\frac{m}{2}$  arestas, isto é, não sabemos determinar a princípio o número mínimo de vezes que o algoritmo randomizado deve ser executado para que se obtenha um corte de arestas de cardinalidade pelo menos  $\frac{m}{2}$ . Para encontrar tal limite consideremos

$$p = \Pr \left( C(A, B) \geq \frac{m}{2} \right)$$

em que  $C(A, B)$  representa a cardinalidade do corte de arestas  $[A, B]$ .

Como sabemos,  $C(A, B) \leq m$  e  $E[C(A, B)] = \frac{m}{2}$ . De acordo com a definição de esperança podemos escrever

$$E[C(A, B)] = \sum_{i \leq \frac{m}{2} - 1} i \cdot \Pr(C(A, B) = i) + \sum_{i \geq \frac{m}{2}} i \cdot \Pr(C(A, B) = i).$$

Temos que

$$\begin{aligned} \sum_{i \leq \frac{m}{2} - 1} i \cdot \Pr(C(A, B) = i) &\leq \left(\frac{m}{2} - 1\right) \cdot \Pr\left(C(A, B) \leq \left(\frac{m}{2} - 1\right)\right) = \\ &= \left(\frac{m}{2} - 1\right) \cdot (1 - p) \end{aligned}$$

e

$$\sum_{i \geq \frac{m}{2}} i \cdot \Pr(C(A, B) = i) \leq m \cdot \Pr\left(C(A, B) \geq \frac{m}{2}\right) = mp.$$

Desse modo

$$\frac{m}{2} = E[C(A, B)] \leq \left(\frac{m}{2} - 1\right) \cdot (1 - p) + mp,$$

o que implica em

$$p \geq \frac{1}{\frac{m}{2} + 1},$$

ou seja, o algoritmo randomizado deverá ser executado pelo menos  $\frac{m}{2} + 1$  vezes para garantir alta probabilidade de obtenção de um corte de arestas com cardinalidade pelo menos  $\frac{m}{2}$ .

Para resolver esse problema de execução, podemos derandomizar esse algoritmo utilizando o método das esperanças condicionais, o qual escolhe uma sequência de decisões de modo a tornar a sequência das esperanças condicionais correspondentes monótona não decrescente. Nesse caso obtemos um algoritmo guloso que parte das mesmas condições iniciais do algoritmo randomizado, mas que a cada etapa posiciona os vértices da sequência em cada parte de modo a maximizar o número de arestas de corte, isto é, se  $|N(v) \cap A| \leq |N(v) \cap B|$ , então ponha  $v$  em  $A$ . Caso contrário ponha  $v$  em  $B$ . Obtemos dessa forma um algoritmo determinístico 2-aproximativo para

SIMPLE MAXCUT. Os detalhes da derandomização podem ser encontrados em [50].

## 1.5 MAXCUT é NP-completo

Em 1971, Stephen Cook introduziu o conceito de problemas de decisão *NP-completos* em seu artigo *The complexity of theorem-proving procedures* [10]. Nesse artigo, Cook provou que o *Problema de Satisfatibilidade Booleana* é NP-completo. Em 1972, partindo dos resultados de Cook, Richard Karp provou no artigo *Reducibility among combinatorial problems* [48] que 21 problemas combinatórios são NP-completos através de sucessivas reduções a partir do problema de Satisfatibilidade Booleana. Entre esses 21 problemas encontrava-se o Problema do Corte Máximo (MAXCUT) com pesos nas arestas.

Alguns anos mais tarde, em 1976, Garey, Johnson e Stockmeyer [40] mostraram que SIMPLE MAXCUT (sem pesos) é NP-completo com base no artigo de Cook e Karp. Desde então, SIMPLE MAXCUT tem sido amplamente estudado na busca de algoritmos aproximativos polinomiais que determinem cortes de arestas com a maior cardinalidade possível, ou, em certos casos, considerando-se subclasses especiais de grafos, algoritmos que determinem exatamente um corte máximo.

Em 1973, Edwards [23, 24] mostrou que em um grafo conexo sempre existe um corte de arestas de tamanho  $\frac{m}{2} + \frac{n-1}{4}$ . Esse fato nos permite concluir que em qualquer grafo com  $n$  vértices,  $m$  arestas e  $t$  componentes conexas, sempre existe um corte de tamanho  $\frac{m}{2} + \frac{n-t}{4}$ , o que melhora o resultado anteriormente obtido por Erdős [26]. A cota inferior  $\frac{m}{2} + \frac{n-1}{4}$  é conhecida como *Edwards-Erdős bound* e esse limite é atingido quando determinamos cortes máximos em grafos completos  $K_n$  com  $n$  ímpar. Demonstrou-se, por várias vezes, em diferentes artigos [7, 14, 27, 53, 66], que esse valor é uma cota inferior para o tamanho do corte máximo de um grafo conexo e, em algumas provas encontram-se algoritmos de tempo polinomial que fornecem um corte com pelo menos  $\frac{m}{2} + \frac{n-1}{4}$  arestas. Por exemplo, no artigo de Ngoc e Tuza [66] de 1993, foi determinado um algoritmo polinomial de tempo  $O(m)$  que encontra um subgrafo bipartido de um grafo  $G$  com pelo menos  $\frac{m}{2} + \frac{n-1}{4}$  arestas.

## 1.6 Casos especiais

Em 1972, Orlova e Dorfman [52], e posteriormente, em 1975, Hadlock [45] mostraram que existe um algoritmo polinomial para determinar o corte máximo em grafos planares. Sua estratégia foi remover o menor número de arestas possível de modo a obter um grafo bipartido, ou seja, eliminar todos os ciclos ímpares do grafo original. Para tal intento, Hadlock considerou o dual  $G_D$  de um grafo planar  $G$  e utilizando as técnicas que resolvem o *Problema do Carteiro Chinês* [69], reduziu o problema do

corte máximo simples à determinação de um emparelhamento de peso máximo no grafo completo formado apenas pelos vértices de grau ímpar de  $G_D$ . A partir desse emparelhamento perfeito de peso máximo pode-se determinar, usando-se o algoritmo polinomial de Edmonds, o número mínimo de arestas que ao serem removidas de  $G$  dão origem a um grafo bipartido.

Em 1979, Garey e Johnson [38] escreveram o primeiro livro abordando exclusivamente a teoria de NP-completude e intratabilidade computacional. Esse livro apresenta um apêndice contendo uma lista com mais de 300 problemas NP-completos agrupados em áreas afins. Dentre esses problemas podemos encontrar MAXCUT na seção A2.2 que fala sobre cortes e conectividade. Com o desenvolvimento do estudo sobre a teoria de NP-completude e a classificação de novos problemas, os autores foram, ao longo dos anos, atualizando a lista de problemas NP-completos conhecidos.

Dado um grafo  $G = (V, E)$  e um conjunto de arestas  $F \subseteq E$ , definimos o *vetor de incidência* de  $F$  por  $x^F \in \mathbb{R}^m$  tal que  $x_e^F = 1$  se  $e \in F$  e  $x_e^F = 0$  caso contrário. O politopo definido por  $P_B(G) = \text{conv}\{x^F \in \mathbb{R}^m | (V, F) \text{ é um subgrafo bipartido de } G\}$  é chamado *politopo dos subgrafos bipartidos de  $G$* . O grafo  $G$  é dito *fracamente bipartido* se  $P_B(G)$  é o conjunto dos vetores  $x \in \mathbb{R}^m$  tais que  $0 \leq x_e \leq 1$  e para todo ciclo ímpar de  $G$  vale a desigualdade  $|x^C| = \sum_{e \in C} x_e \leq |C| - 1$ .

Em 1981, Grötschel e Pulleyblank [44] mostraram que SIMPLE MAXCUT é solúvel em tempo polinomial para grafos fracamente bipartidos. Essa classe de grafos contém os grafos bipartidos e os grafos planares. O algoritmo polinomial apresentado no artigo é baseado no método do elipsóide e em um algoritmo que computa o caminho de comprimento par mais curto. Na mesma época, Barahona [1] mostrou que o problema é solúvel em tempo polinomial para grafos de gênero (genus) fixo.

Em 1984, Grötschel e Nemhauser [43] mostraram que SIMPLE MAXCUT é solúvel em tempo polinomial para grafos cujo ciclo ímpar mais longo não ultrapassa comprimento  $2K + 1$ , em que  $K$  é um inteiro não negativo independente do número de vértices de  $G$ .

Um ano mais tarde, em 1985, Johnson [47] fez a última atualização dos problemas NP-completos conhecidos à época, montando uma tabela na qual listava os resultados em termos de complexidade para vários problemas combinatórios considerados em classes de grafos específicas. Nessa tabela, conhecida como *Coluna Johnson* é possível observar os resultados em termos de complexidade para SIMPLE MAXCUT considerado em algumas dessas classes. Um fato notável é que este problema é NP-completo para a maioria das classes de grafos conhecidas.

Em 1995, Goemans e Williamson [42] encontraram um algoritmo aproximativo polinomial para determinar o corte máximo usando programação semidefinida com razão de aproximação 0,878. Este é o melhor algoritmo aproximativo polinomial encontrado até o presente momento.

Em 2000, Bodlaender e Jansen [6] analisaram o problema do corte máximo simples em algumas classes de grafos, mostrando que esse problema é NP-completo para grafos cordais, split, tripartidos e cobipartidos. Por outro lado, mostraram também que o problema é polinomial para cografos e grafos com treewidth limitada.

Desde então, há um esforço continuado na busca de algoritmos aproximativos polinomiais que forneçam boas soluções para o problema nas classes em que MAX CUT é NP-completo.

A seguir, com o objetivo de deixar o trabalho auto-contido, apresentamos a redução feita por Karp [48] para mostrar que MAXCUT é NP-completo.

## 1.7 O Trabalho de Karp

Karp mostrou que MAX CUT é NP-completo através de uma redução de PARTIÇÃO. A seguir definimos esses problemas na forma de decisão e mostramos como foi construída a redução utilizada por Karp.

### PARTIÇÃO

INSTÂNCIA: Um conjunto finito de inteiros  $C = \{c_1, c_2, \dots, c_n\}$ .

QUESTÃO: Existe um subconjunto  $I$  de  $\{1, 2, \dots, n\}$  tal que  $\sum_{h \in I} c_h = \sum_{h \in C \setminus I} c_h$  ?

### MAX CUT

INSTÂNCIA: Um grafo simples  $G = (V, E)$ , uma função peso  $w : E \rightarrow \mathbb{N}$  e um inteiro positivo  $W$ .

QUESTÃO: Existe um subconjunto  $S \subset V$  tal que  $c(\partial S) \geq W$  ?

Analizando o problema de partição, podemos observar que, se a resposta é SIM, isto é, se existe uma partição do conjunto  $C$  em duas partes disjuntas cujas somas dos elementos são iguais, então cada parte tem soma igual à metade da soma de todos os elementos de  $C$ . De fato, se chamarmos  $s$  a soma de todos os elementos do conjunto  $C$ , teremos a relação  $c_1 + c_2 + \dots + c_n = s$ . E como as partes têm a mesma soma, valem as igualdades  $c_{i_1} + c_{i_2} + \dots + c_{i_p} = c_{i_{p+1}} + c_{i_{p+2}} + \dots + c_{i_n} = A$ , em que  $\{c_{i_1}, c_{i_2}, \dots, c_{i_p}\} = I$  e  $\{c_{i_{p+1}}, c_{i_{p+2}}, \dots, c_{i_n}\} = C \setminus I$ . Lembrando que a soma dos elementos das duas partes recupera a soma de todos os elementos do conjunto  $C$ , teremos a igualdade  $2A = s$  o que leva ao resultado  $A = \frac{s}{2}$ . Portanto pode-se concluir que o problema de partição só terá solução se  $\frac{s}{2}$  for um número inteiro. Caso contrário a resposta ao problema é NÃO, ou seja, não se pode determinar uma partição do conjunto  $C$  em duas partes que possuam a mesma soma.

Dada uma instância  $C$  do problema NP-completo PARTIÇÃO, Karp construiu em tempo polinomial o grafo completo  $G = (V, E)$  com pesos  $w$  do seguinte modo:  $V$  é

o conjunto formado pelos índices associados aos elementos do conjunto  $C$ , ou seja,  $V = \{1, 2, 3, \dots, n\}$  e a função peso  $w$  associada a cada aresta  $\{j, k\} \in E$  é definida por  $w(\{j, k\}) = c_j \cdot c_k$ , e definimos  $W = \left(\frac{\sum_{i=1}^n c_i}{2}\right)^2$ .

Vamos utilizar a notação  $V_1 = \{i_1, i_2, \dots, i_p\}$  e  $V_2 = \{i_{p+1}, i_{p+2}, \dots, i_n\}$  por motivo de simplificação, para representar uma partição de  $V$ .

Afirmamos que o problema da partição de  $C$  tem solução se, e somente se, existe um corte de arestas  $[V_1, V_2]$  de  $G$  satisfazendo a desigualdade

$$\sum_{j \in V_1, k \in V_2} w(j, k) \geq \lceil s^2/4 \rceil.$$

De fato, se o problema da partição de  $C$  tem solução, valem as igualdades  $c_{i_1} + c_{i_2} + \dots + c_{i_p} = \frac{s}{2}$  e  $c_{i_{p+1}} + c_{i_{p+2}} + \dots + c_{i_n} = \frac{s}{2}$  que multiplicadas resultam em  $\sum_{j \in V_1, k \in V_2} c_j \cdot c_k = \frac{s^2}{4}$ , ou seja, a condição  $\sum_{j \in V_1, k \in V_2} w(j, k) \geq \frac{s^2}{4}$  é satisfeita.

Por outro lado, se o problema da partição de  $C$  não tem solução, uma das partes tem soma menor que  $\frac{s}{2}$ , enquanto a outra tem soma maior que este valor. Chamemos  $x = c_{i_1} + c_{i_2} + \dots + c_{i_p}$  e  $y = c_{i_{p+1}} + c_{i_{p+2}} + \dots + c_{i_n}$ . Então o produto dessas igualdades resulta na relação  $xy = \sum_{j \in V_1, k \in V_2} c_j \cdot c_k$ . Lembrando que  $x + y = s$ ,

podemos reescrever a relação anterior como  $x \cdot (s - x) = \sum_{j \in V_1, k \in V_2} c_j \cdot c_k$ , ou seja,

a soma dos pesos associados às arestas do corte  $(V_1, V_2)$  de  $G$  é determinada pela função quadrática  $sx - x^2$  cujo máximo  $\frac{s^2}{4}$  é atingido apenas para  $x = \frac{s}{2}$ . Desse modo, para qualquer outro valor da variável  $x$  obtém-se soma  $(sx - x^2)$  dos pesos do corte estritamente menor do que  $\lceil \frac{s^2}{4} \rceil$ .

Logo o problema da partição de  $C$  tem solução se, e somente se, existe um corte de arestas  $[V_1, V_2]$  de  $G$  satisfazendo a desigualdade  $\sum_{j \in V_1, k \in V_2} w(j, k) \geq W$ .

## 1.8 O Problema do Corte Máximo Simples

Como comentamos acima, o problema do corte máximo pode ser simplificado, considerando-se todas as arestas com igual peso, que pode ser tomado unitário. Nesse caso, o que nos interessa é a cardinalidade do corte de arestas encontrado, ou seja, o corte máximo é aquele que apresenta o maior número de arestas e o problema do corte máximo simples (SIMPLE MAX CUT) pode ser formulado como um problema de decisão na seguinte forma:

### SIMPLE MAX CUT

INSTÂNCIA: Um grafo simples  $G = (V, E)$  e um inteiro positivo  $k$ .

QUESTÃO: Existe um subconjunto  $S \subset V$  tal que o número de arestas com um extremo em  $S$  e o outro em  $V \setminus S$  é maior ou igual a  $k$ ?

Em 1976, Garey, Johnson e Stockmeyer [40] mostraram que SIMPLE MAXCUT é NP-completo através de uma redução de MAX2SAT. Posteriormente, em 1995, Poljak e Tuza [55] mostraram que SIMPLE MAXCUT é NP-completo usando uma redução de STABLE SET. A seguir apresentamos a demonstração de Garey, Johnson e Stockmeyer [40]. Para isso, definimos o problema MAX2SAT.

### MAX2SAT

INSTÂNCIA:  $I = (U, C)$  na qual  $U = \{x_1, x_2, \dots, x_n\}$  é um conjunto de variáveis lógicas,  $C = \{C_1, C_2, \dots, C_p\}$  é um conjunto de cláusulas disjuntivas sobre  $U$  tal que  $|C_i| \leq 2$  para cada  $1 \leq i \leq p$  e um inteiro  $k \geq 0$ .

QUESTÃO: Existe uma atribuição lógica para os literais que satisfaça pelo menos  $k$  cláusulas?

**Teorema 1.3.** [40] SIMPLE MAX CUT é NP-completo.

*Demonstração.* Consideremos as cláusulas  $C_1, C_2, \dots, C_p$  e o inteiro positivo  $k$  dados na instância de MAX2SAT. Sem perda de generalidade, podemos assumir que cada cláusula é formada por dois literais, não necessariamente distintos, e vamos rotulá-las por  $(a_1 \vee b_1), (a_2 \vee b_2), \dots, (a_p \vee b_p)$ , respectivamente. Além disso, podemos assumir que não há duas cláusulas idênticas, pois, caso isso ocorresse, bastaria substituirmos a cláusula  $(u \vee v)$  que se repete pelas cláusulas  $(u \vee c)$  e  $(v \vee \bar{c})$ . Desse modo, se existissem  $q$  cláusulas  $C'_1, C'_2, \dots, C'_q$  aparecendo como cópias de outras já existentes, bastaria trocarmos cada cláusula  $C'_i = (u_i \vee v_i)$  por duas novas cláusulas  $(u_i \vee c_i)$  e  $(v_i \vee \bar{c}_i)$ , em que cada  $c_i$  é uma nova variável, para cada  $i \in \{1, \dots, q\}$  e o inteiro positivo  $k$  seria substituído por  $k + q$  na nova instância de MAX2SAT.

A partir da instância de MAX2SAT construímos uma instância  $G = (V, E)$  de SIMPLE MAXCUT em duas etapas: primeiro exibimos o conjunto de vértices  $V$  e uma estrutura básica de arestas  $E_1$ ; em seguida acrescentamos um conjunto de arestas específicas  $E_2$  que resolvem problemas adicionais, de modo que  $E = E_1 \cup E_2$ .

Sejam  $x_1, x_2, \dots, x_n$  as variáveis presentes nas  $p$  cláusulas. O grafo  $G = (V, E)$

tem conjunto de vértices dado por:

$$\begin{aligned}
V = & \{T_i; 0 \leq i \leq 3p\} \cup \{F_i; 0 \leq i \leq 3p\} \cup \\
& \cup \{t_{ij}; 1 \leq i \leq n, 0 \leq j \leq 3p\} \cup \\
& \cup \{f_{ij}; 1 \leq i \leq n, 0 \leq j \leq 3p\} \cup \\
& \cup \{x_i; 1 \leq i \leq n\} \cup \{\bar{x}_i; 1 \leq i \leq n\}.
\end{aligned}$$

A estrutura básica de arestas  $E_1$  é:

$$\begin{aligned}
E_1 = & \{\{T_i, F_j\}; 0 \leq i \leq 3p, 0 \leq j \leq 3p\} \cup \\
& \cup \{\{t_{ij}, f_{ij}\}; 1 \leq i \leq n, 0 \leq j \leq 3p\} \cup \\
& \cup \{\{x_i, f_{ij}\}; 1 \leq i \leq n, 0 \leq j \leq 3p\} \cup \\
& \cup \{\{\bar{x}_i, t_{ij}\}; 1 \leq i \leq n, 0 \leq j \leq 3p\}.
\end{aligned}$$

Para qualquer partição  $V = V_1 \cup V_2$  dada, dizemos que a aresta  $\{u, v\}$  é *ruim* se ambos os extremos  $u$  e  $v$  pertencem à mesma parte da partição. Caso contrário, dizemos que  $\{u, v\}$  é *boa*.

Se considerarmos uma partição de  $V$  na qual:

- Todos os  $T_i$  pertencem à mesma parte da partição enquanto todos os  $F_i$  pertencem à outra parte e
- Para cada  $i$  os vértices  $x_i$  e  $t_{ij}$  pertencem à mesma parte da partição enquanto  $\bar{x}_i$  e  $f_{ij}$  pertencem à outra parte,

então todas as arestas de  $E_1$  são *boas*. Além disso, se os vértices  $F_i$  e  $F_j$  pertencem a partes distintas da partição, então pelo menos  $3p + 1$  arestas de  $E_1$  são *ruins*, uma vez que os vértices  $F_i$  e  $F_j$  são mutuamente adjacentes a  $3p + 1$  outros vértices (note que os vértices  $T_i$  e  $F_j$  induzem um grafo bipartido completo segundo  $E_1$ ). De modo análogo, se os vértices  $x_i$  e  $\bar{x}_i$  pertencem à mesma parte da partição, então pelo menos  $3p + 1$  arestas de  $E_1$  são *ruins*, já que existem  $3p + 1$  caminhos de comprimento 3 disjuntos em arestas ligando  $x_i$  a  $\bar{x}_i$  (a saber, caminhos da forma  $(x_i, f_{ij}, t_{ij}, \bar{x}_i)$ ).

Nesse ponto vamos acrescentar o conjunto formado pelas arestas adicionais  $E_2$ :

$$\begin{aligned}
E_2 = & \{\{a_i, b_i\}; 1 \leq i \leq p, a_i \neq b_i\} \cup \\
& \cup \{\{a_i, F_{2i-1}\}; 1 \leq i \leq p\} \cup \\
& \cup \{\{b_i, F_{2i}\}; 1 \leq i \leq p\},
\end{aligned}$$

no qual  $a_i$  e  $b_i$  representam os literais presentes na cláusula  $C_i$ .

A instância de SIMPLE MAXCUT é dada pelo grafo  $G = (V, E)$  e pelo inteiro  $W = |E_1| + 2k$ . A Figura 1.4 ilustra o grafo  $G$  correspondente à instância  $I = (U, C)$  de MAX 2-SAT na qual  $U = \{x_1, x_2\}$  e  $C = \{(x_1, \bar{x}_2); (\bar{x}_1, x_2)\}$ . Nessa figura, as arestas de  $E_1$  aparecem com linha cheia, enquanto as de  $E_2$  estão representadas com linhas pontilhadas.

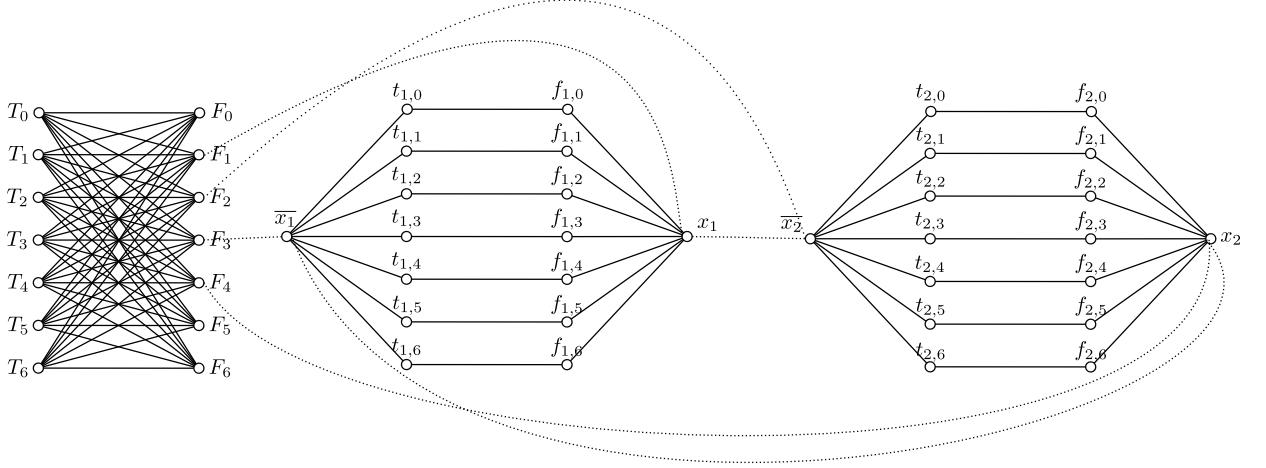


Figura 1.4: Grafo  $G$  associado ao conjunto  $C = \{(x_1, \bar{x}_2); (\bar{x}_1, x_2)\}$ . As arestas com linha cheia pertencem a  $E_1$ , enquanto as pontilhadas formam o conjunto  $E_2$ .

Dada uma atribuição de valores-verdade para as  $n$  variáveis satisfazendo pelo menos  $k$  cláusulas, construa a partição  $V = (V_1, V_2)$  como segue:

$$\begin{aligned} V_1 &= \{F_i; 0 \leq i \leq 3p\} \cup \{x_i; x_i \text{ é falsa}, 1 \leq i \leq n\} \cup \\ &\cup \{t_{ij}; x_i \text{ é falsa}, 1 \leq i \leq n, 0 \leq j \leq 3p\} \cup \\ &\cup \{\bar{x}_i; x_i \text{ é verdadeira}, 1 \leq i \leq n\} \cup \\ &\cup \{f_{ij}; x_i \text{ é verdadeira}, 1 \leq i \leq n, 0 \leq j \leq 3p\}, \\ V_2 &= V \setminus V_1. \end{aligned}$$

A Figura 1.5 ilustra a partição correspondente à atribuição que torna as variáveis  $x_1$  e  $x_2$  falsas. Observe que exatamente quatro arestas de  $E_2$  estão presentes no corte.

Uma vez que, para cada cláusula satisfeita, ou  $a_i$  ou  $b_i$  ou ambos pertencem a  $V_2$ , exatamente duas arestas de  $E_2$  provenientes dessa cláusula devem ser *boas* (a saber, se  $a_i$  é verdadeiro e  $b_i$  falso, teremos as arestas  $\{a_i, b_i\}$  e  $\{a_i, F_{2i-1}\}$ ; se  $a_i$  é falso e  $b_i$  verdadeiro obteremos as arestas  $\{a_i, b_i\}$  e  $\{b_i, F_{2i}\}$  e finalmente caso ambos sejam verdadeiros obteremos as arestas  $\{a_i, F_{2i-1}\}$  e  $\{b_i, F_{2i}\}$ ). Além do mais, pelas nossas observações anteriores, todas as arestas de  $E_1$  são *boas*. Portanto teremos pelo menos  $W = |E_1| + 2k$  arestas *boas*.

Por outro lado, suponhamos que exista uma partição  $V = V_1 \cup V_2$  para a qual

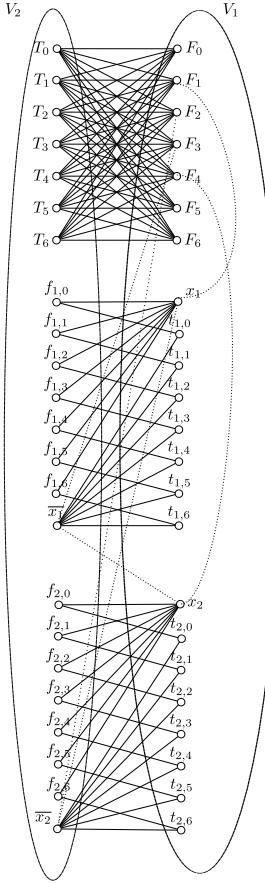


Figura 1.5: Partição correspondente à atribuição  $x_1 = F, x_2 = F$ .

$W$  ou mais arestas são *boas*. Uma vez que  $k > 0$  e  $|E_2| \leq 3p$ , o número de arestas *ruins* não pode ser maior que  $3p$  (lembre-se todas as arestas de  $E_1$  são *boas* segundo a partição dada anteriormente e qualquer mudança nessa partição acarretaria uma perda de pelo menos  $3p+1$  arestas *boas*, o que resultaria, mesmo que todas as arestas de  $E_2$  fossem *boas*, em um corte com no máximo  $|E_1|-1$  arestas). Desse modo todos os vértices  $F_i$  devem pertencer à mesma parte da partição de  $V$ , digamos  $V_1$ . Pelo mesmo motivo, exatamente um dos vértices de cada par  $(x_i, \bar{x}_i)$  deve pertencer a  $V_1$ . Com isso, conseguimos uma atribuição de valores-verdade consistente pondo  $x_i$  verdadeiro se, e somente se,  $x_i \in V_2$ . Para essa atribuição de valores-verdade, a cláusula  $C_i$  é satisfeita toda vez que  $a_i$  ou  $b_i$  ou ambos os vértices pertencem a  $V_2$ . E já observamos que quando isso acontece exatamente duas arestas correspondentes à cláusula  $C_i$  em  $E_2$  são *boas*. Do mesmo modo, não é difícil ver que nenhuma aresta de  $E_2$  correspondente à cláusula  $C_i$  será boa caso ambos os vértices pertençam a  $V_1$ . Consequentemente, como devemos ter  $2k$  arestas de  $E_2$  *boas*, então a atribuição de valores-verdade deve satisfazer pelo menos  $k$  cláusulas.  $\square$

Em 1978, Yannakakis [70] mostrou que SIMPLE MAXCUT é NP-completo para grafos com grau máximo igual a três.

Ao longo dos próximos capítulos desenvolvemos alguns estudos particulares sobre

SIMPLE MAX CUT, com vistas à análise de complexidade do problema restrito a certas classes de grafos, bem como determinação de algoritmos aproximativos polinomiais para estas classes. Consideramos também o problema parametrizado associado e a determinação da complexidade parametrizada do mesmo.

# Capítulo 2

## Complexidade do Problema do Corte Máximo para Grafos Fortemente Cordais

Neste capítulo consideraremos a classe dos grafos fortemente cordais e mostraremos que para esta classe o problema SIMPLE MAX CUT é NP-completo, resolvendo um dos problemas em aberto da coluna Johnson [47].

### 2.1 Grafos Cordais

Um grafo é dito *cordal* ou *triangularizado* quando todo ciclo de comprimento maior que três possui uma *corda*, ou seja, uma aresta unindo dois vértices que não são consecutivos no ciclo. Na Figura 2.1 o grafo de Hajós é cordal, enquanto a roda  $W_6$  não é cordal, pois possui um ciclo externo de comprimento 6 sem nenhuma corda.

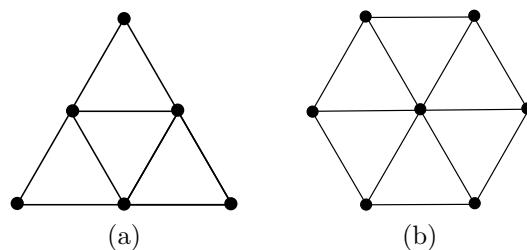


Figura 2.1: (a) Um grafo cordal (grafo de Hajós) e (b) um não cordal (roda  $W_6$ ).

#### 2.1.1 Caracterização e Reconhecimento

Gavril [41] provou que uma das definições alternativas para grafo cordal é a seguinte: um grafo  $G = (V, E)$  é *cordal* se, e somente se, existe uma árvore  $T = (W, F)$  tal que pode-se associar a cada vértice  $v \in V(G)$ , uma subárvore  $T_v = (W_v, E_v)$  de  $T$ , com

$(v, w) \in E$  se, e somente se,  $W_v \cap W_w \neq \emptyset$ . Com base nesta definição, Gavril mostrou que se  $G$  é cordal, então todas as cliques maxima de  $G$  podem ser arrumadas em uma árvore  $T(G)$ , conhecida como *árvore de cliques* ou *cliquetree* tal que para todo vértice  $v \in V(G)$ , as cliques que contêm  $v$  formam uma subárvore de  $T(G)$ .

É importante ressaltar que a classe dos grafos cordais é hereditária, ou seja, todo subgrafo induzido de um grafo cordal é também cordal e essa propriedade nos permite encontrar um algoritmo de reconhecimento para grafos cordais. Pode-se determinar de um modo eficiente se um grafo  $G$  é cordal apresentando um *esquema de eliminação perfeita* para  $G$ . Para definirmos o que vem a ser um esquema de eliminação perfeita, precisamos do conceito de *vértice simplicial*. Um vértice  $v \in V(G)$  é dito *simplicial* se  $N(v)$  é uma clique em  $G$ . Equivalentemente podemos dizer que  $N[v]$  é uma clique em  $G$ . Como exemplo, observe que todo vértice em um grafo completo  $K_n$  é simplicial. Um *Esquema de Eliminação Perfeita (EEP)* de um grafo  $G$  é uma ordenação  $(v_1, v_2, \dots, v_n)$  dos vértices de  $G$  de modo que cada vértice  $v_i$  é simplicial no subgrafo induzido pelos vértices  $\{v_i, \dots, v_n\}$ . Em outras palavras, a sequência  $(v_1, v_2, \dots, v_n)$  é um Esquema de Eliminação Perfeita se, para cada  $v_i$ , o subgrafo induzido por  $N(v_i) - \{v_1, \dots, v_{i-1}\}$  for completo [64].

A seguir citamos dois resultados importantes no sentido do reconhecimento de grafos cordais, o primeiro deles devido a Dirac [19] e o segundo devido a Fulkerson e Gross [36].

**Teorema 2.1.** [19] *Todo grafo cordal que não é uma clique contém pelo menos dois vértices simpliciais não adjacentes.*

**Teorema 2.2.** [36] *Um grafo  $G$  é cordal se, e somente se, possui um Esquema de Eliminação Perfeita.*

A Figura 2.2 ilustra um grafo cordal  $G$  com um Esquema de Eliminação Perfeita.

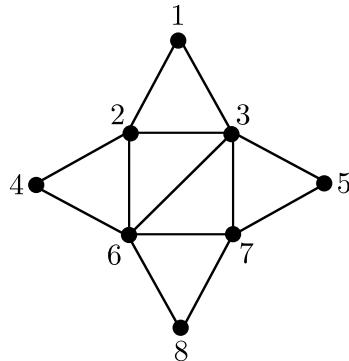


Figura 2.2: Um grafo cordal que possui um esquema de eliminação perfeita  $(1, 4, 5, 8, 2, 7, 3, 6)$ .

Observe que no grafo da Figura 2.2 a vizinhança de cada vértice da ordenação dada induz uma clique sobre o grafo formado apenas pelos vértices que o sucedem

nesta ordenação. De fato,  $N(1) = \{2, 3\}$  é uma clique de  $G$ ,  $N(4) = \{2, 6\}$  é uma clique de  $G[2, \dots, 8]$ ,  $N(5) = \{3, 7\}$  é uma clique de  $G[2, 3, 6, 7, 8]$ , e assim sucessivamente até esgotarmos todos os vértices de  $G$ .

Uma maneira eficiente de obter um esquema de eliminação perfeita em um grafo cordal é através da eliminação sucessiva de seus vértices simpliciais até que todos os vértices do grafo sejam esgotados. Esse processo dá origem a um algoritmo no qual cada vértice simplicial vai sendo removido um a um, de modo que, após a remoção do vértice simplicial  $v$ , o grafo  $G$  é atualizado para  $G := G - \{v\}$ . Caso o grafo  $G$  seja cordal o processo termina quando todos os vértices são removidos. Caso contrário o processo para e informa-se que o grafo dado não é cordal. Se  $G$  é cordal, podemos determinar um Esquema de Eliminação Perfeita para  $G$  listando um a um os vértices simpliciais que foram removidos sucessivamente até esgotar  $V(G)$ .

A Figura 2.3 apresenta a eliminação sucessiva dos vértices simpliciais que deu origem ao Esquema de Eliminação Perfeita do grafo  $G$  apresentado na Figura 2.2.

Pode-se encontrar um Esquema de Eliminação Perfeita para um grafo cordal através de um algoritmo de tempo linear que consiste de duas etapas. Numa primeira etapa, faz-se uso do algoritmo de Busca Lexicográfica [59] ou do algoritmo MCS(maximum cardinality search) [65] para computar uma ordenação e, numa segunda etapa, verifica-se se essa ordenação é um Esquema de Eliminação Perfeita.

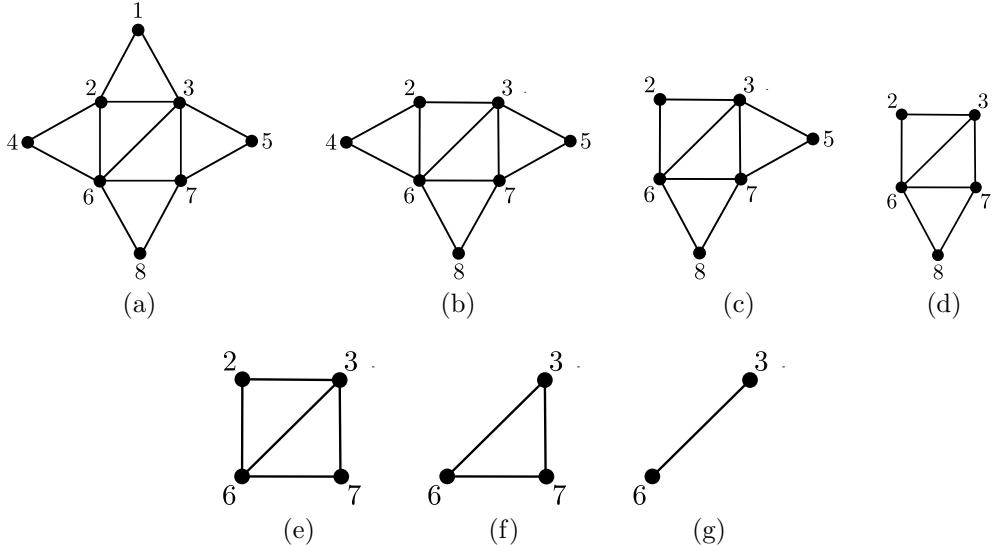


Figura 2.3: (a) Ilustração do esquema de eliminação perfeita  $(1, 4, 5, 8, 2, 7, 3, 6)$  do grafo da Figura 2.2: (b),(c),(d),(e),(f),(g).

## 2.2 Grafos Fortemente Cordais

Um grafo  $G = (V, E)$  é *fortemente cordal* se  $G$  é cordal e todo ciclo de comprimento par maior ou igual a seis tem uma *corda ímpar*, isto é, uma aresta que conecta dois

vértices não consecutivos no ciclo que estão a uma distância ímpar um do outro sobre este ciclo. No item (a) da Figura 2.4 temos um grafo não cordal com uma corda ímpar  $\{v_2, v_5\}$  e no item (b) um grafo fortemente cordal.

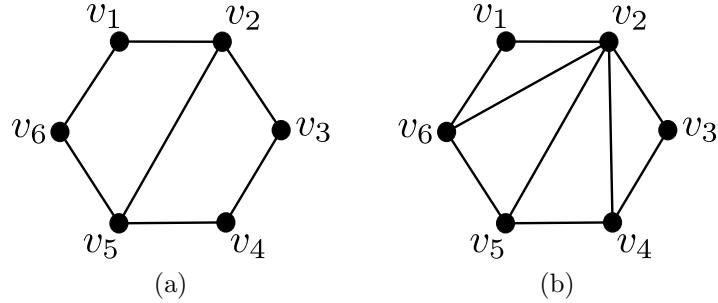


Figura 2.4: (a) Grafo não cordal com uma corda ímpar e (b) Grafo fortemente cordal.

Um  $n$ -sol ou  $n$ -trampolim é um grafo cordal que possui um ciclo hamiltoniano  $(y_1, x_1, y_2, x_2, \dots, y_n, x_n, y_1)$ , ( $n \geq 3$ ) no qual cada  $x_i$  tem grau dois, isto é, o conjunto de vértices pode ser particionado em dois subconjuntos  $X = \{x_1, \dots, x_n\}$  e  $Y = \{y_1, \dots, y_n\}$  tais que cada vértice  $x_i$  em  $X$  tem exatamente dois vizinhos  $y_i$  e  $y_{(i+1) \bmod n}$ . É fácil ver que um sol não pode ser um grafo fortemente cordal pois o ciclo hamiltoniano presente nesse grafo não possui nenhuma corda ímpar. Quando o conjunto  $\{y_1, y_2, \dots, y_n\}$  forma uma clique, dizemos que o  $n$ -sol é *completo*. Caso contrário, dizemos que o  $n$ -sol é *incompleto*. Na Figura 2.5 damos exemplos de alguns  $n$ -sóis.

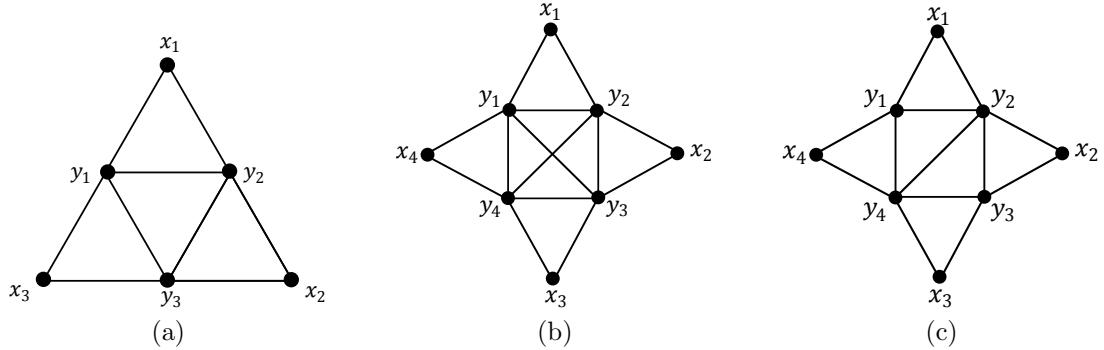


Figura 2.5: (a) Um 3-sol completo (grafo de Hajós), (b) um 4-sol completo e (c) um 4-sol incompleto.

Grafos fortemente cordais podem ser caracterizados através de subgrafos proibidos: um grafo  $G$  é *fortemente cordal* se, e somente se,  $G$  é um grafo cordal que não possui nenhum  $n$ -sol *completo* como subgrafo induzido [29].

Grafos fortemente cordais também podem ser caracterizados por possuírem um *esquema de eliminação forte*. Um *Esquema de Eliminação Forte (EEF)* de um grafo  $G$  é uma ordenação  $(v_1, v_2, \dots, v_n)$  dos vértices de  $G$  satisfazendo às duas condições seguintes para cada  $i, j, k, l$  em  $\{1, 2, \dots, n\}$  com  $i > j > k > l$ :

(i) Se  $(v_i, v_k), (v_j, v_k) \in E(G)$ , então  $(v_i, v_j) \in E(G)$ ;

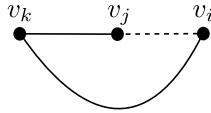


Figura 2.6: condição (i).

(ii) Se  $(v_i, v_l), (v_j, v_l), (v_j, v_k) \in E(G)$ , então  $(v_i, v_k) \in E(G)$ .

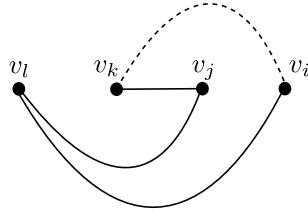


Figura 2.7: condição (ii).

### 2.2.1 Caracterização e Reconhecimento

Em seu artigo de 1983, Farber [29] determinou que uma condição necessária e suficiente para um grafo cordal ser fortemente cordal é a existência de um esquema de eliminação forte para esse grafo. Com essa condição é possível produzir um algoritmo de reconhecimento para grafos fortemente cordais em tempo polinomial. A seguir enunciamos dois teoremas devidos a Farber que permitem a construção do algoritmo de reconhecimento para grafos fortemente cordais em tempo polinomial.

Com o objetivo de construir de modo simples um Esquema de Eliminação Forte para um grafo  $G = (V, E)$ , introduziremos algumas definições.

Dois vértices  $u$  e  $v$  de um grafo  $G$  são *compatíveis* se  $N[u] \subset N[v]$  ou  $N[v] \subset N[u]$ . Usamos a simbologia  $u \sim v$  para dizer que os vértices  $u$  e  $v$  são compatíveis. Caso contrário dizemos que  $u$  e  $v$  são *incompatíveis*.

Um vértice  $x \in V(G)$  é *simple* se os vértices de  $N[v]$  são dois a dois compatíveis, ou seja, se o conjunto  $\{N[u] | u \in N[v]\}$  formado pelas vizinhanças fechadas dos vértices de  $N[v]$  pode ser ordenado linearmente por inclusão.

**Teorema 2.3.** [29] *Todo grafo fortemente cordal não trivial tem pelo menos dois vértices simples.*

**Teorema 2.4.** [29] *Um grafo é fortemente cordal se, e somente se, possui um esquema de eliminação forte.*

A Figura 2.8 apresenta um exemplo de um grafo fortemente cordal e um esquema de eliminação forte para esse grafo. Podemos observar que existem dois vértices

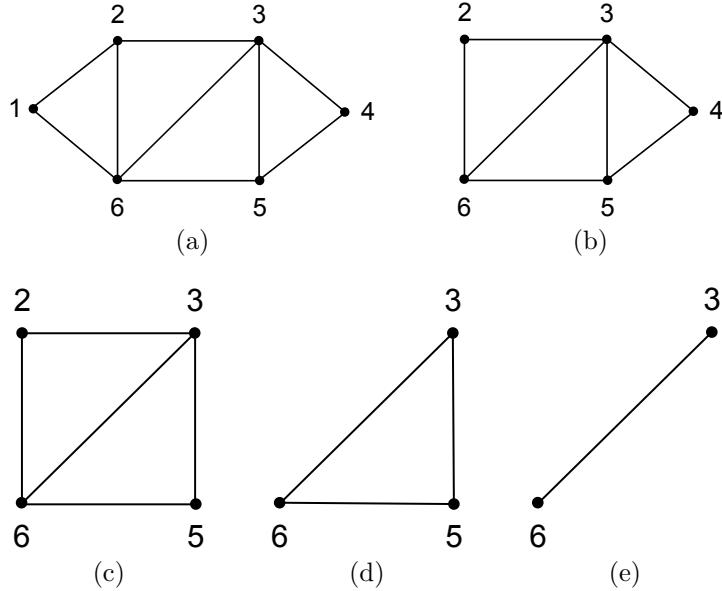


Figura 2.8: (a) Um grafo fortemente cordal e (b),(c),(d),(e) um esquema de eliminação forte  $(6, 3, 5, 2, 4, 1)$  para esse grafo.

simples, a saber 1 e 4. De fato,  $N(1) = \{2, 6\}$  e tem-se  $N[2] = \{1, 2, 3, 6\} \subset \{1, 2, 3, 5, 6\} = N[6]$ , do mesmo modo que  $N(4) = \{3, 5\}$  e  $N[3] \supset N[5]$ . Eliminando o vértice 1 obtemos um novo grafo fortemente cordal no qual o vértice 4 continua sendo simples pois preserva a mesma vizinhança  $\{3, 5\}$  e as vizinhanças fechadas dos vértices 3 e 5 ainda são aninhadas como anteriormente. Eliminamos então o vértice 4, obtendo um novo grafo fortemente cordal com dois vértices simples, a saber 2 e 5, ambos com mesma vizinhança  $N(2) = N(5) = \{3, 6\}$  e  $N[3] = N[6] = \{2, 3, 5, 6\}$ . Eliminando o vértice 2 obtemos um grafo  $K_3$  com todos os vértices simples, isto é, a ordem de eliminação a esta altura é indiferente e podemos eliminar sequencialmente 5, 3 e 6, obtendo a sequência  $(1, 4, 2, 5, 3, 6)$  de eliminação dos vértices simples. Observe que a sequência reversa fornece um esquema de eliminação forte para o grafo, isto é, a ordenação  $(6, 3, 5, 2, 4, 1)$  é um esquema de eliminação forte para o grafo dado e esse fato nos garante que o grafo é fortemente cordal.

Observe que o procedimento descrito acima nos fornece um algoritmo linear para o reconhecimento de um grafo fortemente cordal. De fato, iniciamos com um grafo  $G$  e procuramos um vértice simples  $v \in V(G)$ . Feito isso, eliminamos  $v$  atualizando o grafo, de modo que  $G := G - \{v\}$ . Buscamos um vértice simples no grafo atualizado, eliminamos esse vértice e o processo continua recursivamente. Se em algum momento o grafo atualizado não apresenta nenhum vértice simples, o algoritmo retorna que  $G$  não é fortemente cordal. Caso contrário, esgotamos todos os vértices de  $G$ , obtendo uma sequência de eliminação simples, que revertida produz um esquema de eliminação forte para  $G$ , provando que  $G$  é fortemente cordal.

## 2.3 SIMPLE MAX CUT em Grafos Fortemente Cordais

Lembramos aqui que Bodlaender e Jansen [6] provaram que SIMPLE MAX CUT é NP-completo para grafos cordais em 2000. Nossa contribuição consiste em observar que a instância de Bodlaender e Jansen [6] é um grafo cordal que é também fortemente cordal. Utilizando a redução feita por Bodlaender e Jansen [6], mostramos que SIMPLE MAX CUT é NP-completo para grafos fortemente cordais a partir do problema MAX 2-SAT, o qual apresentamos a seguir em sua forma de decisão.

**Teorema 2.5.** SIMPLE MAX CUT é NP-completo para grafos fortemente cordais.

*Demonstração.* O problema MAX CUT está em NP, uma vez que pode-se verificar em tempo polinomial se  $|(S, V \setminus S)| \geq k$ .

Bodlaender e Jansen apresentaram uma transformação polinomial de MAX 2-SAT para SIMPLE MAX CUT em grafos cordais utilizando a seguinte construção.

Seja  $X = \{x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$  o conjunto de literais de  $U$  e  $(a_1 \vee b_1), \dots, (a_p \vee b_p)$  o conjunto de cláusulas da instância de MAX 2-SAT, em que as literais  $a_i$  e  $b_j$  pertencem ao conjunto  $X$ .

Consideremos  $m = 2p$ . Vamos definir os seguintes conjuntos com o objetivo de construir o grafo  $G' = (V', E')$  que compõe a instância de SIMPLE MAX CUT :

1.  $C^{(i)} = \{c_1^{(i)}, \dots, c_{m+1}^{(i)}\}$ ,  $D^{(i)} = \{d_1^{(i)}, \dots, d_{m+2}^{(i)}\}$ ,  $E^{(i)} = \{e_1^{(i)}, \dots, e_{m+2}^{(i)}\}$ , para cada  $i \in \{1, \dots, n\}$ .
2.  $T^{(i)} = \{t_1^{(i)}, \dots, t_{m+2}^{(i)}\}$ , para cada  $i \in \{1, \dots, n\}$ , e  $R = \{r_1, \dots, r_p\}$ ,  $Q = \{q_1, \dots, q_p\}$ ,  $Y = \{y_1, \dots, y_p\}$  e  $S = \{s_1, \dots, s_{m+1}\}$ .
3.  $U = \{u_1, \dots, u_p\}$ ,  $V = \{v_1, \dots, v_p\}$ ,  $W = \{w_1, \dots, w_p\}$  e  $Z = \{z_1, \dots, z_p\}$ .

A seguir definimos o grafo  $G' = (V', E')$  anteriormente citado e uma partição  $(V_1, V_2)$  do conjunto de vértices  $V'$  na qual  $V_1$  contem os literais com valor-verdade *falso* e  $V_2$  contem os literais com valor-verdade *verdadeiro*. O conjunto de vértices  $V'$  é a união disjunta de todos os conjuntos mencionados anteriormente, isto é,  $C^{(i)} (1 \leq i \leq n), D^{(i)} (1 \leq i \leq n), E^{(i)} (1 \leq i \leq n), S, R, Q, T^{(i)} (1 \leq i \leq p), U, V, W, Y, Z$ . Existe uma aresta entre cada par de vértices de  $V'$  se, e somente se, ambos os vértices pertencerem a pelo menos um dos seguintes conjuntos, ou seja, cada um dos seguintes conjuntos forma uma clique em  $G'$ :

- para cada  $i \in \{1, \dots, n\}$ :

–  $\{x_i\} \cup C^{(i)} \cup D^{(i)}$ ,

–  $\{x_i\} \cup C^{(i)} \cup E^{(i)}$ ,

–  $\{x_i, \bar{x}_i\} \cup C^{(i)}$ ;

- para cada  $j \in \{1, \dots, m+1\}$ , temos a clique  $R \cup \{s_j\}$ ;
- $X \cup R \cup Y$ ;
- para cada  $i \in \{1, \dots, p\}$ , e cada  $j \in \{1, \dots, m+2\}$ , temos a clique  $\{r_i, q_i, t_j^{(i)}\}$ ;
- para cada  $i \in \{1, \dots, p\}$ , se a  $i$  – éssima cláusula é  $(a_i \vee b_i)$ , então temos as cliques:
  - $\{a_i, b_i, r_i, q_i\}$ ,
  - $\{a_i, b_i, v_i, w_i\}$ ,
  - $\{a_i, u_i, v_i\}$ ,
  - $\{b_i, w_i, z_i\}$ .

O grafo  $G'$  é cordal porque podemos arrumar todas as cliques anteriormente definidas em uma árvore  $T$ , tal que todo vértice pertence a um conjunto de árvores que formam uma subárvore conexa de  $T$ . Veja a Figura 2.9. Mais do que isso, provaremos a seguir que  $G'$  é fortemente cordal, exibindo um esquema de eliminação forte para  $G'$ .

### 2.3.1 Esquema de eliminação forte de $G'$

A seguir descrevemos um esquema de eliminação forte considerando que em cada etapa uma sequência de conjuntos de vértices simples serão removidos do grafo original, definindo um novo grafo para a etapa seguinte. Observe que a cada etapa as vizinhanças dos vértices remanescentes no grafo são atualizadas pela remoção dos vértices eliminados nas etapas anteriores.

#### 1. Eliminando os vértices dos conjuntos $D^{(i)}$ :

Para cada  $i \in \{1, \dots, n\}$ , os vértices  $d_j^{(i)}$  pertencem unicamente à clique  $\{x_i\} \cup C^{(i)} \cup D^{(i)}$ . Portanto temos  $N[d_j^{(i)}] = \{x_i\} \cup C^{(i)} \cup D^{(i)}$ . Com o objetivo de simplificar a notação, escreveremos  $N[D^{(i)}] = \{x_i\} \cup C^{(i)} \cup D^{(i)}$  significando que para todo vértice  $d_j^{(i)} \in D^{(i)}$ , tem-se  $N[d_j^{(i)}] = \{x_i\} \cup C^{(i)} \cup D^{(i)}$ . Se  $x_i = a_j$  para algum  $j \in \{1, 2, \dots, p\}$ , então  $N[x_i] = X \cup R \cup Y \cup C^{(i)} \cup D^{(i)} \cup E^{(i)} \cup \{v_j, w_j, u_j, q_j\}$ . Se  $x_i = b_k$  para algum  $k \in \{1, 2, \dots, p\}$ , então  $N[x_i] = X \cup R \cup Y \cup C^{(i)} \cup D^{(i)} \cup E^{(i)} \cup \{v_k, w_k, z_k, q_k\}$ . Caso contrário, se  $x_i$  não é literal de nenhuma cláusula, então  $N[x_i] = X \cup R \cup Y \cup C^{(i)} \cup D^{(i)} \cup E^{(i)}$ .

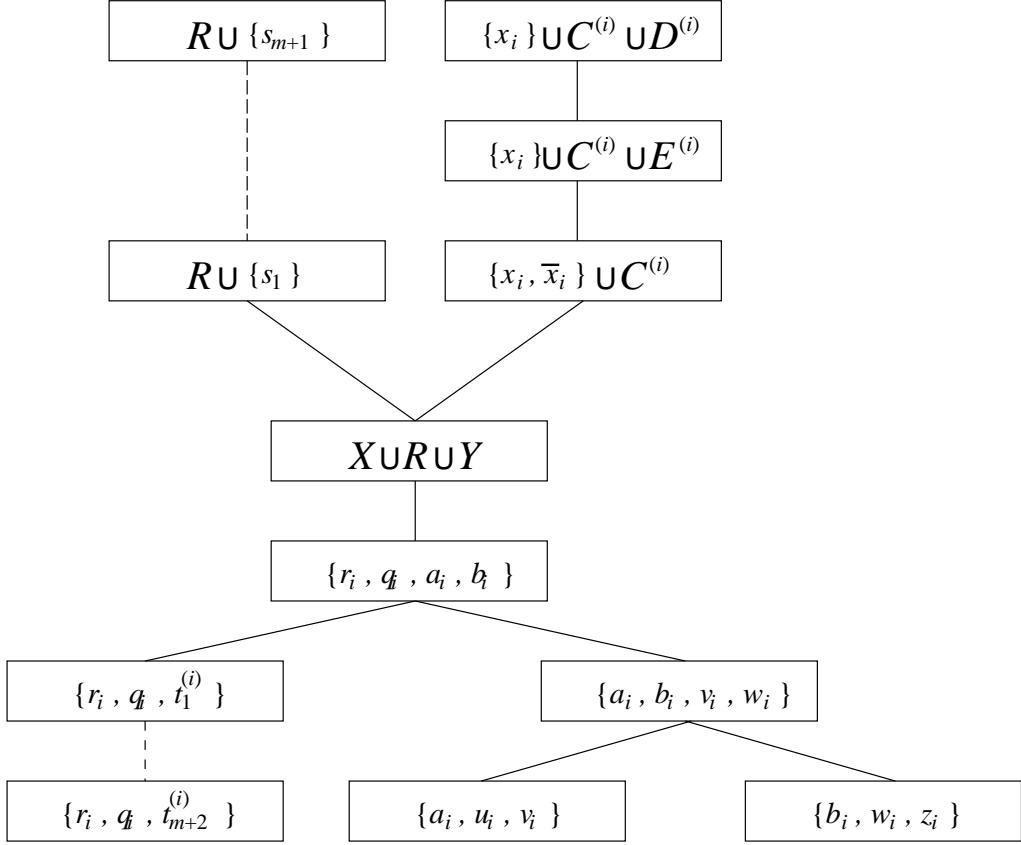


Figura 2.9: Árvore de cliques de  $G'$ .

E  $N[C^{(i)}] = \{x_i, \bar{x}_i\} \cup C^{(i)} \cup D^{(i)} \cup E^{(i)}$ . Portanto podemos ver que todos os vértices de  $D^{(i)}$  são *simples* para cada  $i \in \{1, \dots, n\}$ , o que nos permite eliminá-los.

2. Eliminando os vértices dos conjuntos  $E^{(i)}$ :

Analiamos os conjuntos  $E^{(i)}$  para cada  $i \in \{1, \dots, n\}$ ,  $N[E^{(i)}] = \{x_i\} \cup C^{(i)} \cup E^{(i)}$ . Se  $x_i = a_j$  para algum  $j \in \{1, 2, \dots, p\}$ , então  $N[x_i] = X \cup R \cup Y \cup C^{(i)} \cup E^{(i)} \cup \{v_j, w_j, u_j, q_j\}$ . Se  $x_i = b_k$  para algum  $k \in \{1, 2, \dots, p\}$ , então  $N[x_i] = X \cup R \cup Y \cup C^{(i)} \cup E^{(i)} \cup \{v_k, w_k, z_k, q_k\}$ . Caso contrário, se  $x_i$  não é literal de nenhuma cláusula, então  $N[x_i] = X \cup R \cup Y \cup C^{(i)} \cup E^{(i)}$ . E  $N[C^{(i)}] = \{x_i, \bar{x}_i\} \cup C^{(i)} \cup E^{(i)}$ . Novamente podemos ver que os vértices de  $E^{(i)}$  são simples, o que nos permite eliminar todos os vértices de  $E^{(i)}$ , para cada  $i \in \{1, \dots, n\}$ .

3. Eliminando os vértices do conjunto  $U$ :

Para cada  $i \in \{1, \dots, p\}$ , temos  $N[u_i] = \{u_i, a_i, v_i\}$ ,  $N[v_i] = \{a_i, b_i, v_i, u_i, w_i\}$  e  $N[a_i] = X \cup R \cup Y \cup \{q_i, u_i, v_i, w_i\}$ . Concluimos que todos os vértices de  $U$  são simples e os eliminamos.

4. Eliminando os vértices do conjunto  $Z$ :

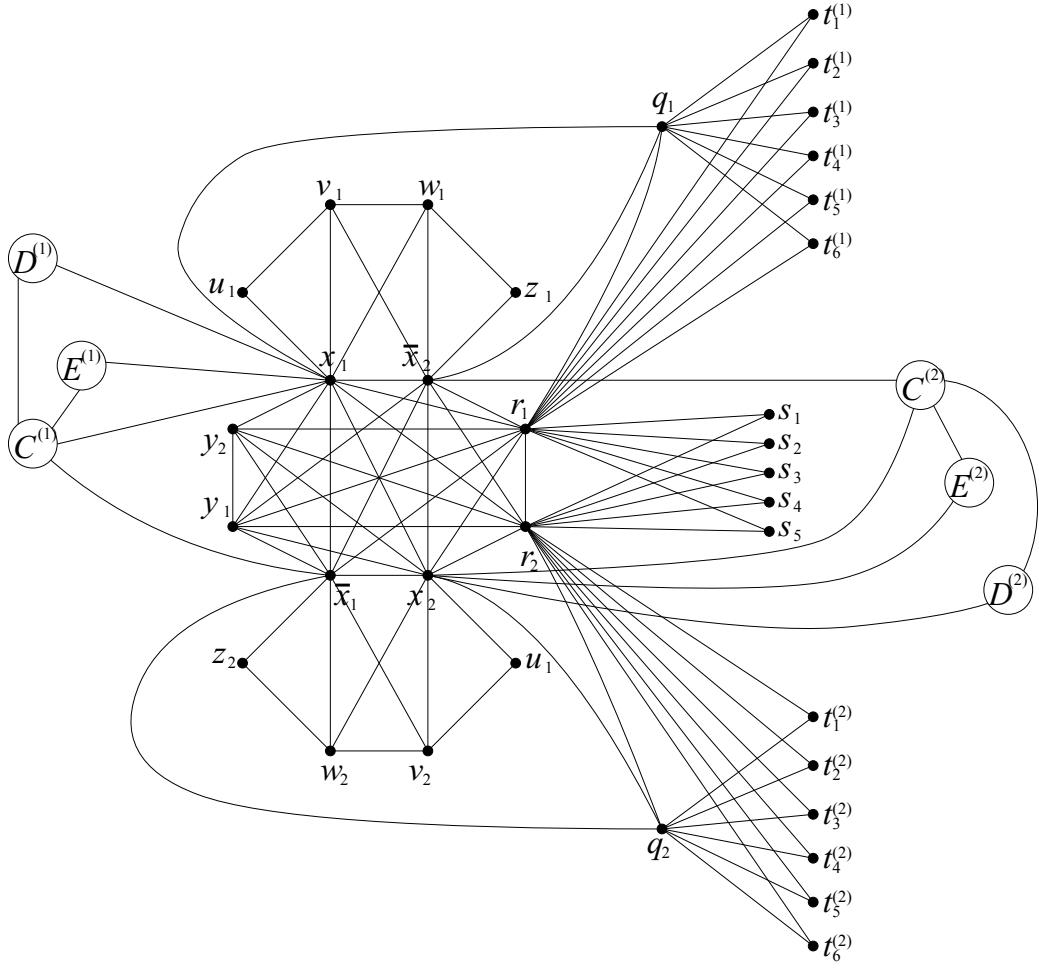


Figura 2.10: Grafo  $G'$  obtido a partir da instância  $I = (X, C) = (\{x_1, x_2, \bar{x}_1, \bar{x}_2\}, \{(x_1 \vee \bar{x}_2), (x_2 \vee \bar{x}_1)\})$  de MAX-2-SAT.

Agora consideramos os vértices do conjunto  $Z$ . Temos  $N[z_i] = \{b_i, w_i, z_i\}$ ,  $N[b_i] = X \cup R \cup Y \cup \{v_i, w_i, z_i, q_i\}$  e  $N[w_i] = \{a_i, b_i, v_i, w_i, z_i\}$ . Então, todos os vértices de  $Z$  são simples e eliminamos os vértices de  $Z$ .

5. Eliminando os vértices do conjunto  $V$ :

Podemos ver que  $N[v_i] = \{a_i, b_i, v_i, w_i\}$ ,  $N[a_i] = N[b_i] = X \cup R \cup Y \cup \{v_i, w_i, q_i\}$  e  $N[w_i] = \{a_i, b_i, v_i, w_i\}$ . Logo concluimos que todos os vértices de  $V$  são simples e eliminamos todos os vértices de  $V$ .

6. Eliminando os vértices do conjunto  $W$ :

Temos  $N[w_i] = \{a_i, b_i, w_i\}$ ,  $N[a_i] = N[b_i] = X \cup R \cup Y \cup \{q_i, w_i\}$ . Desse modo todos os vértices de  $W$  são simples e vamos eliminar todos os vértices de  $W$ .

7. Eliminando os vértices dos conjuntos  $T^{(i)}$ :

Para cada  $i \in \{1, \dots, n\}$ , podemos ver que  $N[t_j^{(i)}] = \{t_j^{(i)}, r_i, q_i\}$ ,  $N[q_i] = \{r_i, q_i, a_i, b_i\} \cup T^{(i)}$  e  $N[r_i] = \{q_i\} \cup T^{(i)} \cup R \cup X \cup Y \cup S$ . Portanto, todo vértice

de  $T^{(i)}$  é simples, para cada  $i \in \{1, \dots, p\}$ , e eliminamos todos os vértices de  $T^{(i)}$ ,  $i \in \{1, 2, 3, \dots, p\}$ .

8. Eliminando os vértices do conjunto  $Q$ :

Temos  $N[q_i] = \{r_i, q_i, a_i, b_i\}$ ,  $N[r_i] = \{q_i\} \cup R \cup X \cup Y \cup S$  e  $N[a_i] = N[b_i] = X \cup R \cup Y \cup \{q_i\}$ . Consequentemente todos os vértices de  $Q$  são simples e eliminamos todos os vértices de  $Q$ .

9. Eliminando os vértices do conjunto  $S$ :

Temos  $N[s_i] = \{s_i\} \cup R$  e  $N[r_j] = X \cup Y \cup R \cup S$ . Portanto todos os vértices de  $S$  são simples e eliminamos todos os vértices de  $S$ .

10. Eliminando os vértices dos conjuntos  $C^{(i)}$ :

Temos  $N[C^{(i)}] = \{x_i, \bar{x}_i\} \cup C^{(i)}$  e  $N[x_i] = N[\bar{x}_i] = X \cup Y \cup R \cup C^{(i)}$ . Portanto todos os vértices de  $C^{(i)}$  são simples e eliminamos todos os vértices de  $C^{(i)}$ .

11. Eliminando os vértices do conjunto  $R \cup X \cup Y$ :

Podemos ver que  $N[R] = N[X] = N[Y] = X \cup Y \cup R$ . Consequentemente todos esses três conjuntos são formados por vértices simples e eliminamos todos os vértices de  $R \cup X \cup Y$ .

Isso conclui o esquema de eliminação forte. A Figura 2.11 ilustra o esquema de eliminação forte para o grafo  $G'$  da Figura 2.10.

□

Os resultados obtidos nesse capítulo encontram-se no artigo “*A Complexidade do Problema do Corte Máximo para Grafos Fortemente Cordais*”, que foi apresentado no XLV SBPO, 2013, e publicado nos Anais do XLV SBPO, 2013 [62].

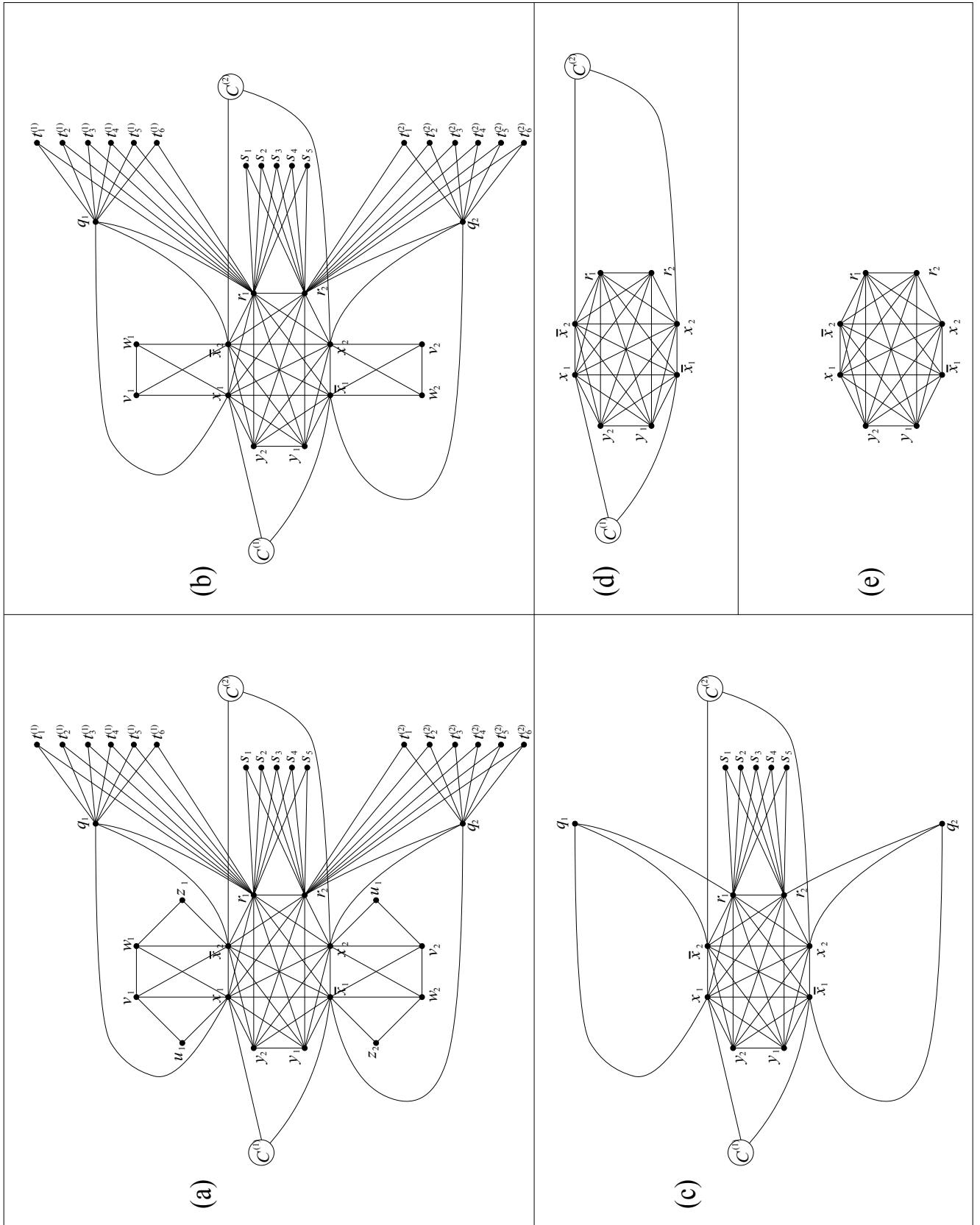


Figura 2.11: Esquema de eliminação forte para o grafo fortemente cordal  $G'$ : (a) Eliminação dos conjuntos  $D^{(i)}$  e  $E^{(i)}$ ; (b) Eliminação dos conjuntos  $U$  e  $Z$ ; (c) Eliminação dos conjuntos  $V$ ,  $W$  e  $T^{(i)}$ ; (d) Eliminação dos conjuntos  $Q$  e  $S$ ; (e) Eliminação dos conjuntos  $C^{(i)}$  e posterior eliminação do clique  $X \cup Y \cup R$ .

# Capítulo 3

## Algoritmos Aproximativos Polinomiais para o Problema do Corte Máximo em Grafos Split

### 3.1 Definições Preliminares

Um grafo  $G = (V, E)$  é *split* se o conjunto  $V$  dos vértices de  $G$  admite uma partição  $V = K \cup S$  na qual  $K$  é uma *clique* e  $S$  é um *conjunto independente*. Se todos os vértices de  $S$  possuem o mesmo grau  $k$ , diremos que  $G$  é um grafo *k-split*. Se o grafo *k-split*  $G$  possuir uma clique maximal com  $n$  vértices, então diremos que  $G$  é um grafo *(k, n)-split*. Note que devemos ter sempre  $k < n$ , pois de outro modo a clique  $K$  não seria maximal, podendo ser aumentada. Na Figura 3.1 temos exemplos de tais grafos.

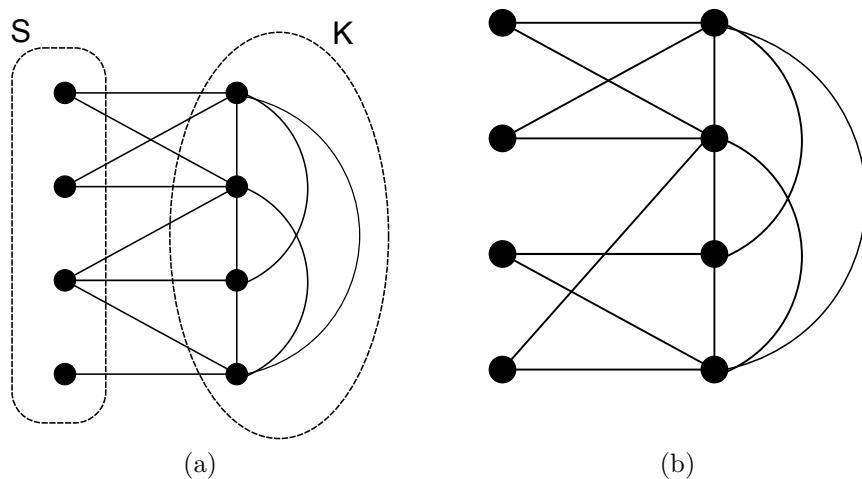


Figura 3.1: (a) Um grafo split e (b) Um grafo  $(2,4)$ -split.

### 3.2 Algoritmos Aproximativos Polinomiais

Como já foi mostrado por Bodlaender e Jansen [6], SIMPLE MAXCUT é NP-completo para grafos 2-split.

Desse modo, vamos apresentar um algoritmo aproximativo para a determinação do Corte Máximo para grafos split, utilizando um resultado de Bodlaender, Figueiredo, Gutierrez, Kloks e Niedermeier [4] juntamente com um lema que fornece uma cota superior para o tamanho do corte máximo de um grafo split.

**Lema 3.1.** [4] *Seja  $K_n$  o grafo completo com  $n$  vértices. Então  $mc(K_n) = \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ . Além disso, qualquer partição de  $V$  em duas partes  $A$  e  $B$  com cardinalidades  $|A| = \lceil \frac{n}{2} \rceil$  e  $|B| = \lfloor \frac{n}{2} \rfloor$  dá origem a um corte máximo de  $K_n$ .*

*Demonstração.* Seja  $(A, B)$  uma partição dos vértices de  $K_n$ , tal que  $|A| = p$  e  $|B| = q$ , ( $p + q = n$ ). Então o corte de arestas  $[A, B]$  é isomorfo a  $K_{p,q}$  o grafo bipartido completo que tem exatamente  $pq = p(n - p)$  arestas. Logo  $|[A, B]| = p(n - p)$  que assume seu valor máximo quando  $p = \lfloor \frac{n}{2} \rfloor$  ou  $p = \lceil \frac{n}{2} \rceil$ . De fato, a função  $f : \mathbb{R} \rightarrow \mathbb{R}$ ,  $f(p) = p(n - p)$  quadrática atinge seu valor máximo no ponto que anula a derivada  $f'(p) = n - 2p$ , isto é, quando  $p = \frac{n}{2}$ . Como estamos considerando apenas valores inteiros para a variável  $p$ , então a função atingirá valor máximo para os valores de  $p$  inteiros mais próximos de  $\frac{n}{2}$  que é a abcissa do vértice da parábola associada a essa função. Portanto temos  $|[A, B]| \leq \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$  e esse valor é atingido quando  $|A| = \lceil \frac{n}{2} \rceil$ .  $\square$

Uma ilustração do lema anterior pode ser vista na Figura 3.2.

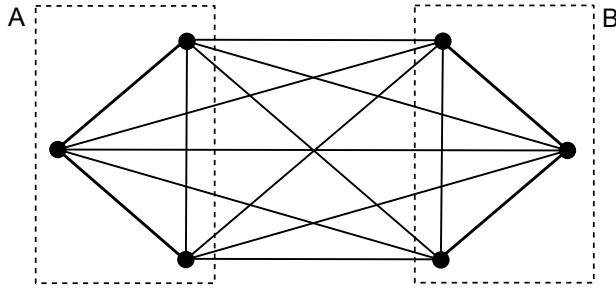


Figura 3.2: Exemplo de corte máximo para o grafo completo  $K_6$ .

**Lema 3.2.** *Se  $G = (V, E)$  é um grafo split com partição  $V = K \cup S$  no qual a clique máxima tem tamanho  $|K| = n$  e  $k = \frac{\sum_{v \in S} d(v)}{|S|}$  é o grau médio do conjunto independente  $S$ , então para todo corte de arestas  $[A, B]$  tem-se  $|[A, B]| \leq \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor + k|S|$ .*

*Demonstração.* Observe que de acordo com o lema anterior, o melhor corte possível para a clique maximal  $K$  tem tamanho  $\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ . Uma partição  $(A, B)$  de  $V$  induz as

partições  $(K_A, K_B)$  em  $K$  e  $(S_A, S_B)$  em  $S$ . Como  $S$  é um conjunto independente, não há arestas entre  $S_A$  e  $S_B$ . Logo, basta considerarmos os seguintes subconjuntos de  $E$ :  $[K_A, K_B]$ ,  $[K_A, S_B]$  e  $[K_B, S_A]$ . Desse modo, podemos decompor o corte de arestas  $[A, B]$  numa união disjunta  $[A, B] = [K_A, K_B] \cup [K_A, S_B] \cup [K_B, S_A]$ . Lembrando que  $[K_A, S_B] \cup [K_B, S_A] \subseteq [K, S]$ , teremos  $|[K_A, S_B]| + |[K_B, S_A]| \leq |[K, S]|$ . Além disso,  $|[K, S]| = \sum_{v \in S} d(v) = k|S|$ . A partir das desigualdades  $|[K_A, K_B]| \leq \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$  e  $|[K_A, S_B]| + |[K_B, S_A]| \leq k|S|$ , podemos concluir que  $|[A, B]| \leq \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor + k|S|$ .  $\square$

### 3.2.1 Um Algoritmo $\frac{2}{3}$ -aproximativo para SIMPLE MAX CUT

Sejam  $G = (V, E)$  um grafo split com partição de vértices  $V = K \cup S$ , na qual  $K$  é a clique máxima com  $|K| = n$ ,  $S$  é o conjunto independente, e  $N(S) = \cup_{v \in S} N(v)$  a vizinhança do conjunto independente  $S$ . A seguir, apresentamos um algoritmo aproximativo que fornece uma solução viável de SIMPLE MAXCUT através da construção de um corte de arestas  $[A, B]$  do grafo  $G$ .

**Caso 1:** Se  $|N(S)| \leq \lfloor \frac{n}{2} \rfloor$ , ponha todos os vértices de  $N(S)$  em  $A$  e, se necessário, complete essa parte com outros vértices de  $K \setminus N(S)$  até obter  $\lfloor \frac{n}{2} \rfloor$  vértices na parte  $A$ . Ponha todos os vértices de  $S$  e os vértices restantes de  $K$  na parte  $B$ .

A Figura 3.3 ilustra essa situação. Veremos adiante no Teorema 3.3 que o corte obtido nesse caso é exatamente o corte máximo.

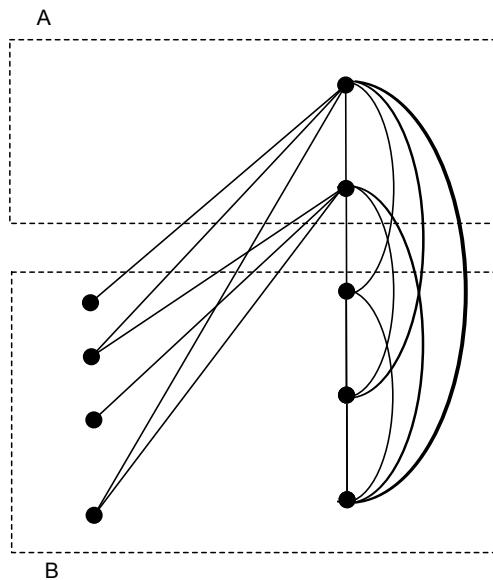


Figura 3.3: Exemplo de corte máximo para um grafo split com  $|N(S)| \leq \lfloor \frac{n}{2} \rfloor$ .

**Caso 2:** Se  $|N(S)| > \lfloor \frac{n}{2} \rfloor$ , calcule o grau médio dos vértices de  $S$ :  $k = \frac{\sum_{v \in S} d(v)}{|S|}$ . Se  $k|S| \leq 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ , então escolha aleatoriamente  $\lfloor \frac{n}{2} \rfloor$  vértices de  $N(S)$  e coloque-os

na parte  $A$ . Ponha os demais vértices de  $K$  na parte  $B$ . Para cada vértice  $v \in S$ , se  $|N(v) \cap A| \geq |N(v) \cap B|$ , ponha  $v$  em  $B$ , caso contrário ponha  $v$  em  $A$ . Se  $k|S| > 2\lceil\frac{n}{2}\rceil\lfloor\frac{n}{2}\rfloor$  faça  $A = S$  e  $B = K$ .

A Figura 3.4 ilustra as duas possibilidades para o segundo caso.

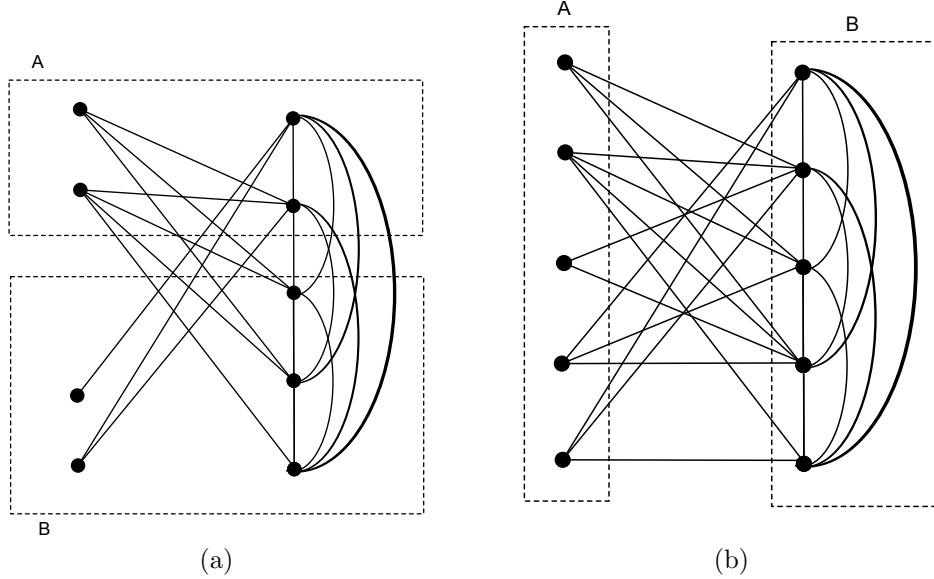


Figura 3.4: Exemplo de corte aproximado para um grafo split com  $|N(S)| \geq \lfloor\frac{n}{2}\rfloor$ : (a)  $k|S| \leq 2\lceil\frac{n}{2}\rceil\lfloor\frac{n}{2}\rfloor$  e (b)  $k|S| > 2\lceil\frac{n}{2}\rceil\lfloor\frac{n}{2}\rfloor$ .

No caso do problema SIMPLE MAXCUT, a razão de aproximação do algoritmo será obtida pelo quociente entre o tamanho do corte fornecido pelo algoritmo dado e o tamanho do corte máximo real. Contudo, como não sabemos determinar o tamanho do corte máximo, vamos substituir esse valor pela cota superior fornecida no Lema 3.2.

Provamos, no teorema seguinte, que o algoritmo aproximativo encontrado tem razão  $\frac{2}{3}$ .

**Teorema 3.3.** *Se  $G = (V, E)$  é um grafo split com partição  $V = K \cup S$ , então o algoritmo aproximativo para SIMPLE MAXCUT tem razão  $\frac{2}{3}$ .*

*Demonstração.* Nas condições do caso 1, isto é, quando  $|N(S)| \leq \lfloor\frac{n}{2}\rfloor$ , o algoritmo fornece exatamente o corte máximo  $\lceil\frac{n}{2}\rceil\lfloor\frac{n}{2}\rfloor + k|S| = |[A, B]|$  para  $G$ . De fato, todas as arestas do conjunto  $[S, K]$  pertencem ao corte, e  $|[S, K]| = k|S|$ . Além disso, pela construção feita, a clique  $K$  foi particionada de modo que, em cada parte há, respectivamente  $\lfloor\frac{n}{2}\rfloor$  e  $\lceil\frac{n}{2}\rceil$  vértices de  $K$ . Então o corte obtido tem tamanho  $\lceil\frac{n}{2}\rceil\lfloor\frac{n}{2}\rfloor + k|S|$  que, pelo Lema 3.2, é o maior possível.

Teremos dois subcasos a considerar para o caso 2, no qual  $|N(S)| > \lfloor\frac{n}{2}\rfloor$ , ou seja, quando o tamanho da vizinhança do conjunto independente  $S$  satisfaz:  $k|S| \leq 2\lceil\frac{n}{2}\rceil\lfloor\frac{n}{2}\rfloor$  ou  $k|S| > 2\lceil\frac{n}{2}\rceil\lfloor\frac{n}{2}\rfloor$ .

Se  $k|S| \leq 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ , podemos observar que a construção dada pelo algoritmo partitiona a clique  $K$  em duas partes que possuem, respectivamente,  $\lfloor \frac{n}{2} \rfloor$  e  $\lceil \frac{n}{2} \rceil$  vértices. Desse modo, com relação a essa clique teremos exatamente  $\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$  arestas de corte. Como  $|N(S)|$  ultrapassa a metade de  $|K|$ , há possivelmente vértices de  $S$  com vizinhos em ambas as partes da partição. Pela construção, posicionamos esses vértices de modo a obter o maior número possível de arestas de corte, isto é, se  $v \in S$  e  $|N(v) \cap A| \geq |N(v) \cap B|$ , colocamos  $v$  em  $B$ , de modo a obter pelo menos metade das arestas com extremo em  $v$  como arestas de corte. Consequentemente isso origina um corte com pelo menos metade das arestas de  $[S, K]$ , isto é, o corte final obtido tem pelo menos  $\frac{k|S|}{2} + \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$  arestas. Então a razão de aproximação do algoritmo será  $\frac{2}{3}$ , uma vez que vale a seguinte cadeia de desigualdades:

$k|S| \leq 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$  implica em  $\frac{k|S|}{2} \leq \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ . Somando aos dois membros dessa desigualdade a expressão  $\frac{3k|S|}{2} + 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$  obteremos:  $\frac{3k|S|}{2} + 3\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor \geq 2k|S| + 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$  a partir da qual podemos concluir diretamente que  $\frac{\frac{k|S|}{2} + \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor}{k|S| + \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor} \geq \frac{2}{3}$ .

Se  $k|S| > 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ , o algoritmo fornece um corte de tamanho  $k|S|$  e, somando  $2k|S|$  aos dois membros da desigualdade anterior, obteremos  $3k|S| > 2k|S| + 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ . Novamente, a partir dessa desigualdade, obtemos  $\frac{k|S|}{k|S| + \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor} > \frac{2}{3}$ .  $\square$

A seguir apresentamos um exemplo de um grafo split no qual a razão de aproximação dada é exatamente  $\frac{2}{3}$ . Considere  $G$  o grafo 3-split no qual  $|K| = 5$  e  $|S| = 4$ . A Figura 3.5 ilustra uma possível partição desse grafo obtida pelo algoritmo approximativo dado anteriormente.

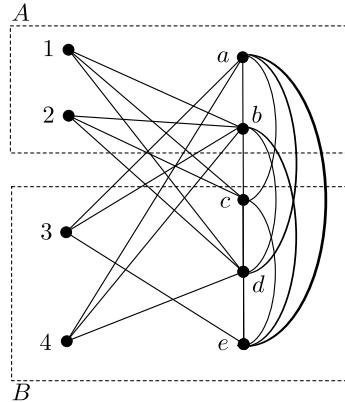


Figura 3.5: Grafo split no qual o algoritmo approximativo tem razão de aproximação  $\frac{2}{3}$ .

### 3.3 Grafos $(k, n)$ -Split Cheios

Definimos a seguir a classe dos grafos  $(k, n)$ -split cheios. Seja  $G$  um grafo  $(k, n)$ -split. Dizemos que  $G = (V, E)$  é  $(k, n)$ -split cheio se  $V = K \cup S$  e, para cada subconjunto  $W$  formado por  $k$  vértices da clique  $K$ , existe um subconjunto não

vazio e maximal  $U \subset S$  tal que cada vértice  $u \in U$  é adjacente a todos os vértices de  $W$ . A *multiplicidade*  $m_S(W)$  de  $W$  em  $S$  é a cardinalidade do conjunto  $U$ , isto é,  $m_S(W) = |U|$ . Se  $m_S(W) = 1$ , para todo  $W \subset K$ , com  $|W| = k$ , dizemos que  $G$  é um *grafo  $(k, n)$ -split cheio simples*. Caso contrário, dizemos que  $G$  é um *grafo  $(k, n)$ -split cheio múltiplo*. A Figura 3.6 mostra exemplos desses grafos.

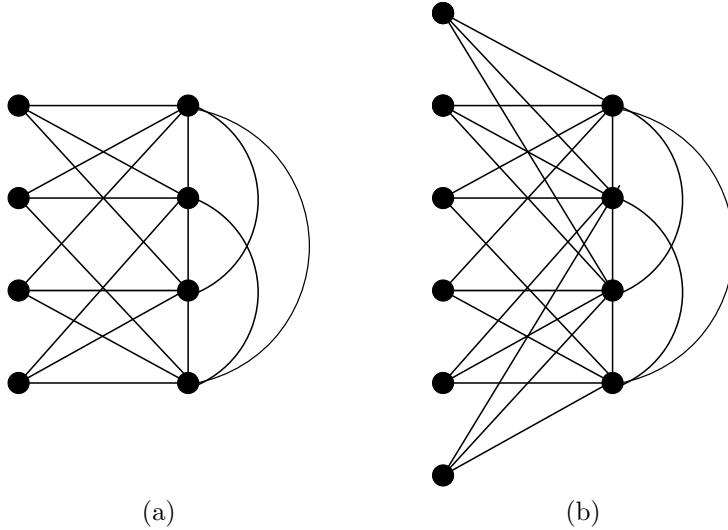


Figura 3.6: (a) Um grafo  $(3,4)$ -split cheio simples e (b) um grafo  $(3,4)$ -split cheio múltiplo.

A seguir determinamos alguns resultados sobre o corte máximo de grafos  $(k, n)$ -split cheios.

**Teorema 3.4.** *Se  $G$  é um grafo  $(1, n)$ -split, então  $mc(G) = |S| + \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ .*

*Demonastração.* Como  $G$  é  $(1, n)$ -split, particionamos a clique  $K$  pondo aleatoriamente  $\lfloor \frac{n}{2} \rfloor$  vértices em  $A$  e  $\lceil \frac{n}{2} \rceil$  vértices em  $B$ . Em seguida, se  $v \in K \cap A$ , ponha  $N_S(v)$  em  $B$ . Caso contrário, ponha  $N_S(v)$  em  $A$ . Dessa forma, todas as arestas de  $[S, K]$  pertencem ao corte e obtemos um corte de tamanho  $\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor + |[S, K]| = \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor + |S|$  que é o limite superior determinado no lema 3.2. Portanto  $mc(G) = |S| + \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ .  $\square$

A Figura 3.7 ilustra um grafo 1-split e seu corte máximo.

Segue imediatamente desse teorema o resultado para grafos  $(1, n)$ -split cheios:

**Corolário 3.5.** *Se  $G$  é um grafo  $(1, n)$ -split cheio, então  $mc(G) = |S| + \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ .*

**Teorema 3.6.** *Se  $G = (S, K)$  é um grafo  $(k, n)$ -split cheio simples com  $k \geq 2$ , então  $mc(G) = k|S|$ .*

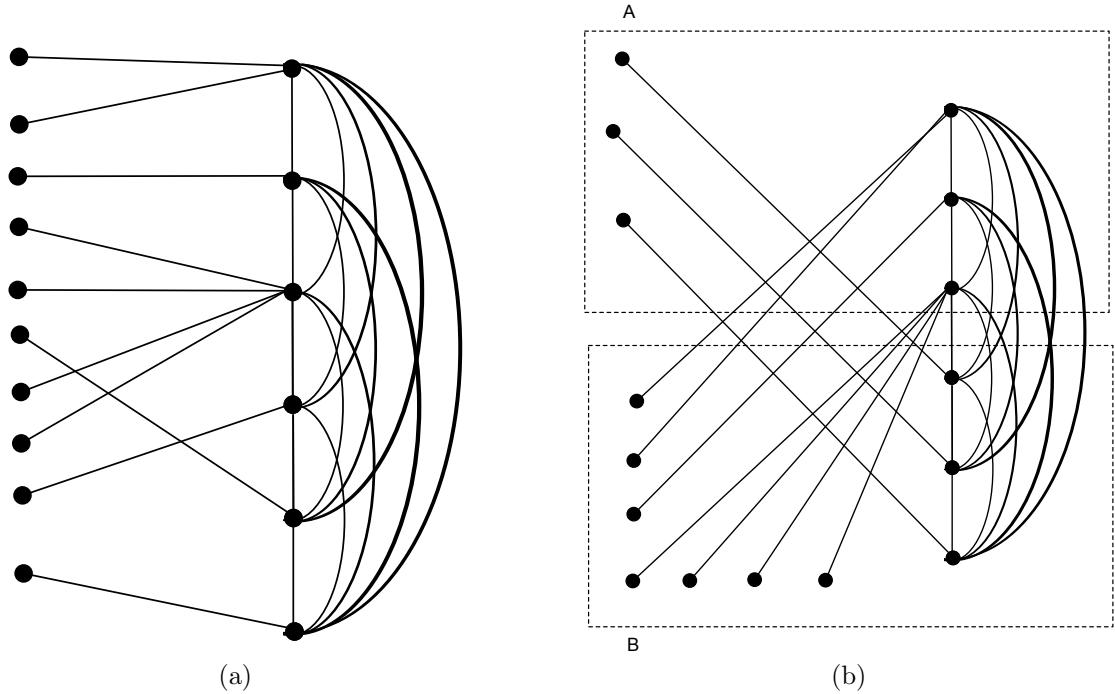


Figura 3.7: (a) Um grafo 1-split e (b) seu corte máximo.

*Demonstração.* Para o estudo deste caso, vamos considerar três subcasos nos quais utilizaremos a seguinte propriedade combinatória: Se um conjunto  $X$  com  $n$  elementos for particionado em dois subconjuntos  $Y$  e  $Z$  com  $p$  e  $q$  elementos, respectivamente, o número de subconjuntos de  $X$  com  $k$  elementos,  $\binom{n}{k}$ , pode ser obtido através da soma dos números de subconjuntos de  $Y$  e de  $Z$  com  $k$  elementos, mais a soma dos números de composições de elementos de  $Y$  e de  $Z$  que resultam em subconjuntos com  $k$  elementos (veja a Figura 3.8). Suponha que  $G$  seja  $(k, n)$ -split cheio simples. Desse modo temos:

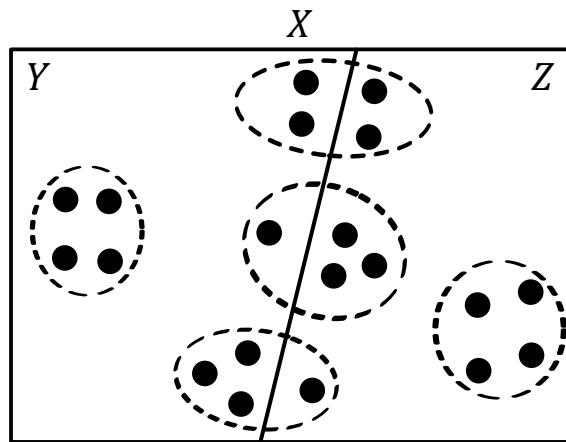


Figura 3.8: Partição de um conjunto  $X$  em duas partes  $Y$  e  $Z$ . Possíveis escolhas de subconjuntos de  $X$  com cardinalidade 4.

**Subcaso 1:**  $k \leq \min\{p, q\}$

$$\binom{n}{k} = \binom{p}{k} + \binom{q}{k} + \sum_{i=1}^{k-1} \binom{p}{i} \binom{q}{k-i}$$

se  $k \leq p$  e  $k \leq q$ . Consequentemente, o número de arestas do corte  $[S, K]$  em um grafo  $(k, n)$ -split cheio simples é dado por

$$k \binom{n}{k} = k \binom{p}{k} + k \binom{q}{k} + \sum_{i=1}^{k-1} k \binom{p}{i} \binom{q}{k-i}$$

sempre que  $k \leq p$  e  $k \leq q$ .

Portanto, se tomarmos uma partição da clique  $K = (A, B)$ , tal que  $|A| = p$  e  $|B| = q$ , podemos posicionar os vértices do conjunto independente  $S$  de modo a obter o maior número possível de arestas de corte, colocando os vértices que são adjacentes somente aos elementos de  $A$  na parte  $B$  e vice-versa, obtendo um total de  $k \binom{p}{k} + k \binom{q}{k}$  arestas de corte. Caso um vértice  $u \in S$  tenha vizinhos em ambas as partes, procedemos do seguinte modo: se  $|N(u) \cap A| \geq |N(u) \cap B|$ , ponha  $u$  em  $B$ , caso contrário ponha  $u$  em  $A$ . Esse procedimento resulta num total de  $\sum_{i=1}^{k-1} \max(i, k-i) \binom{p}{i} \binom{q}{k-i}$  arestas de corte.

Logo

$$|[A, B]| = k \binom{p}{k} + k \binom{q}{k} + \sum_{i=1}^{k-1} \max(i, k-i) \binom{p}{i} \binom{q}{k-i} + pq.$$

Observe que, quaisquer que sejam  $p \geq k$  e  $q \geq k$  tem-se  $p \leq \binom{p}{i}$  e  $q \leq \binom{q}{k-i}$  se  $1 \leq i \leq k-1$ . Consequentemente

$$pq \leq \binom{p}{i} \binom{q}{k-i} \leq \sum_{i=1}^{k-1} \binom{p}{i} \binom{q}{k-i}.$$

Note também que  $\max(i, k-i) \leq k-1$  para  $1 \leq i \leq k-1$ . Desse modo, obtemos a seguinte desigualdade:

$$\begin{aligned} |[A, B]| &= k \binom{p}{k} + k \binom{q}{k} + \sum_{i=1}^{k-1} \max(i, k-i) \binom{p}{i} \binom{q}{k-i} + pq \leq \\ &= k \binom{p}{k} + k \binom{q}{k} + \sum_{i=1}^{k-1} (k-1) \binom{p}{i} \binom{q}{k-i} + \sum_{i=1}^{k-1} \binom{p}{i} \binom{q}{k-i} = k \binom{p}{k} + \\ &= k \binom{q}{k} + \sum_{i=1}^{k-1} k \binom{p}{i} \binom{q}{k-i} = k \binom{n}{k} = k|S|. \end{aligned}$$

**Subcaso 2:**  $p < k < q$

Nesse caso, dada uma partição  $(A, B)$  de  $K$  tal que  $|A| = p$  e  $|B| = q$ , o número de elementos de  $S$  será dado pela soma do número de subconjuntos de  $B$  com  $k$  elementos, mais o número de possíveis combinações de elementos de  $A$  e de  $B$  que resultam em conjuntos com  $k$  elementos, isto é,

$$\binom{n}{k} = \binom{q}{k} + \sum_{i=1}^p \binom{p}{i} \binom{q}{k-i},$$

quaisquer que sejam  $p$  e  $q$  inteiros positivos tais que  $p + q = n$  com  $p < k < q$ . Procedendo como no caso anterior, obtemos o corte de arestas  $[A, B]$  de tamanho

$$|[A, B]| = pq + k \binom{q}{k} + \sum_{i=1}^p \max(i, k-i) \binom{p}{i} \binom{q}{k-i}.$$

Como  $p \leq \binom{p}{i}$  e  $q \leq \binom{q}{k-i}$ , segue daí que

$$pq \leq \binom{p}{i} \binom{q}{k-i} \leq \sum_{i=1}^p \binom{p}{i} \binom{q}{k-i}. \quad (3.1)$$

E também  $\max(i, k-i) \leq k-1$ . Logo temos a seguinte desigualdade:

$$\begin{aligned} |[A, B]| &= pq + k \binom{q}{k} + \sum_{i=1}^p \max(i, k-i) \binom{p}{i} \binom{q}{k-i} < \\ &= k \binom{q}{k} + \sum_{i=1}^p (k-1) \binom{p}{i} \binom{q}{k-i} + \sum_{i=1}^p \binom{p}{i} \binom{q}{k-i} = \\ &= k \binom{q}{k} + \sum_{i=1}^p k \binom{p}{i} \binom{q}{k-i} = k \binom{n}{k} = k|S| \end{aligned}$$

**Subcaso 3:**  $p < q < k$

Num último caso, o número de elementos de  $S$  será dado unicamente pelas possíveis combinações entre elementos de  $A$  e de  $B$  que resultam em subconjuntos com  $k$  elementos, isto é,

$$\binom{n}{k} = \sum_{i=k-q}^p \binom{p}{i} \binom{q}{k-i}. \quad (3.2)$$

Desse modo, o corte de arestas  $[A, B]$  terá tamanho

$$|[A, B]| = pq + \sum_{i=k-q}^p \max(i, k-i) \binom{p}{i} \binom{q}{k-i}.$$

A partir de 3.1, 3.2 e da desigualdade  $\max(i, k-i) \leq k-1$  para todo  $k-q \leq i \leq p$ , segue que

$$|[A, B]| \leq k \binom{n}{k} = k|S|.$$

□

Os resultados desse capítulo encontram-se no artigo “*Approximation Algorithms for Simple Max Cut of Split Graphs*” [63] que concorreu ao prêmio Roberto Diéguez Galvão no XLVI SBPO, 2014 e encontra-se publicado nos Anais do XLVI SBPO, 2014.

# Capítulo 4

## Complexidade Parametrizada do Problema do Corte Máximo Sinalizado

Um grafo  $G = (V, E)$  é *sinalizado* se cada aresta for rotulada como *positiva*( $+$ ) ou *negativa*( $-$ ). De outro modo podemos dizer que existe uma função  $c : E \rightarrow \{+, -\}$  que atribui a cada aresta  $e \in E$  um *sinal*  $c(e) \in \{+, -\}$ . A Figura 4.1 mostra um exemplo de grafo sinalizado.

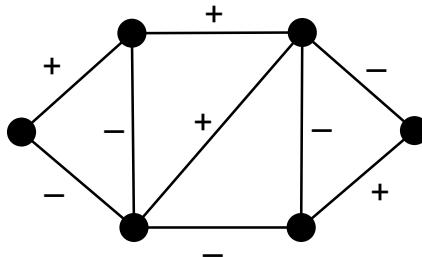


Figura 4.1: Exemplo de um grafo sinalizado.

Seja  $(V_1, V_2)$  uma partição de  $V$ . Dizemos que o grafo  $G$  é  $(V_1, V_2)$ -*balanceado* se os subgrafos induzidos  $G[V_1]$  e  $G[V_2]$  só possuem arestas positivas, enquanto o conjunto  $[V_1, V_2]$  só possui arestas negativas.

Um grafo  $G$  é *balanceado* se existe uma partição  $(V_1, V_2)$  de  $V$  tal que  $G$  é  $(V_1, V_2)$ -balanceado. A Figura 4.2 mostra um grafo sinalizado balanceado e sua partição  $(V_1, V_2)$ . O problema de reconhecimento de grafos sinalizados balanceados é polinomial. Essas definições foram estabelecidas por Harary [46] num artigo de 1953. Neste artigo, Harary provou que um grafo sinalizado é balanceado se, e somente se, não possui ciclos negativos. Um ciclo negativo é um ciclo que apresenta uma quantidade ímpar de arestas negativas. Com essa propriedade é possível construir um algoritmo de tempo polinomial para o reconhecimento de grafos sinalizados balanceados como foi mencionado acima. De fato, basta ordenar cada ciclo e percorrer

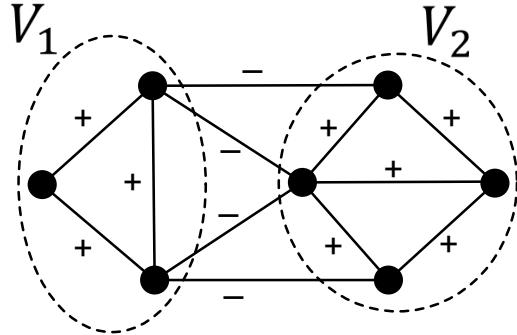


Figura 4.2: Exemplo de um grafo sinalizado balanceado com sua partição.

seus vértices, posicionando-os na mesma parte da partição caso a aresta percorrida tenha sinal positivo. Caso a aresta tenha sinal negativo, posicionamos estes vértices em partes distintas.

Na Figura 4.3 mostramos um grafo sinalizado não balanceado e seu subgrafo balanceado de tamanho máximo.

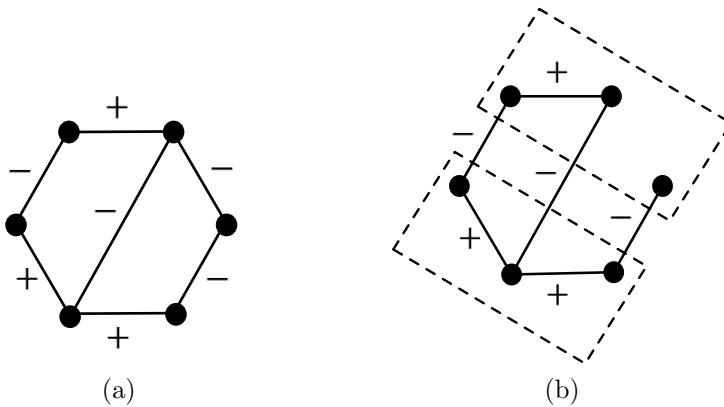


Figura 4.3: (a) Um grafo sinalizado não balanceado e (b) seu subgrafo balanceado máximo.

Dado um grafo sinalizado  $G$ , o problema da determinação do subgrafo balanceado de  $G$  com o maior número possível de arestas é chamado **SIGNED MAX CUT**. Se considerarmos a restrição do problema **SIGNED MAX CUT** apenas às instâncias  $G$  cujas arestas só apresentam sinal negativo, então o problema **SIGNED MAX CUT** se reduz a **SIMPLE MAX CUT**. De fato, basta procurarmos uma partição  $(V_1, V_2)$  dos vértices de  $G$  que forneça um corte de arestas  $[V_1, V_2]$  com a maior cardinalidade possível e eliminarmos as arestas que estão fora do corte. Portanto, **SIGNED MAX CUT** é NP-difícil, uma vez que **SIMPLE MAX CUT** também é [40]. Assim, **SIGNED MAX CUT** é um problema mais geral que **SIMPLE MAX CUT**. Esse fato nos motiva a buscar abordagens alternativas para determinar soluções de **SIGNED MAX CUT**. Uma dessas abordagens é a análise do ponto de vista da Complexidade Parametrizada.

Ao longo desse capítulo, o problema **SIGNED MAX CUT** será abordado sob a

perspectiva da Complexidade Parametrizada. Este capítulo é autocontido, mas citamos [20], [34], [33] e [51] como referências para um estudo introdutório desse assunto.

O objetivo do estudo da complexidade parametrizada de um problema computacional é classificar o problema de acordo com o nível de dificuldade de solução com respeito a um ou mais parâmetros da instância de entrada ou de saída. A complexidade será medida como uma função de tais parâmetros. O *parâmetro* é uma variável escolhida na instância do problema. Sejam  $\Pi$  um problema de decisão,  $G$  o grafo de entrada de  $\Pi$  e  $k$  o parâmetro escolhido sobre  $G$ . Dizemos que  $\Pi(G, k)$  é *tratável a parâmetro fixo (FPT)* quando existe um algoritmo que resolve  $\Pi$  em tempo  $f(k) \cdot n^{O(1)}$  para alguma função computável  $f$ . Isto significa que ao fixarmos o parâmetro  $k$ , obteremos uma solução polinomial para o problema  $\Pi$ . Fellows [21] mostrou que um problema parametrizado  $\Pi(G, k)$  é FPT se, e somente se, admite um *núcleo*. O processo de obtenção do núcleo, conhecido como *kernelização*, bem como a definição desse conceito serão dados na Seção 4.2.

Considerando SIGNED MAX CUT, estamos interessados em determinar uma parametrização desse problema que nos forneça um corte balanceado com pelo menos  $\frac{m}{2} + \frac{n-1}{4}$  arestas, de modo a obter núcleos(*ernels*) com ordem polinomial baixa em certas classes de grafos. De fato, como já mencionamos anteriormente no capítulo 1, Edwards [23] e [24] mostrou que qualquer grafo conexo com  $n$  vértices e  $m$  arestas tem um corte de tamanho  $\frac{m}{2} + \frac{n-1}{4}$  e esse valor é *justo*, pois é realizado em grafos completos de ordem ímpar. A Figura 4.4 ilustra esse fato. Como já foi dito, essa cota inferior para o tamanho do corte de arestas do grafo é conhecida como *Edwards-Erdős bound* e justifica a seguinte parametrização de MAX CUT recentemente considerada por Crowston, Jones e Mnich [16]: Dado um grafo conexo  $G$  e um parâmetro  $k$ , decidir se  $G$  tem um corte de arestas de tamanho maior ou igual a  $\frac{m}{2} + \frac{n-1}{4} + \frac{k}{4}$ . Crowston, Jones e Mnich [16] provaram que este problema é NP-completo, produziram algoritmos FPT para essa parametrização do problema, e determinaram um núcleo de tamanho  $O(k^5)$  para grafos em geral.

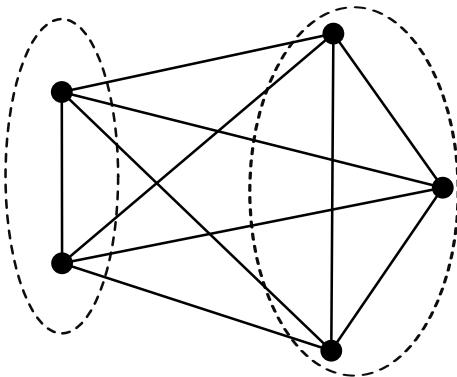


Figura 4.4: Partição de  $V(K_5)$  com  $\frac{m}{2} + \frac{n-1}{4} = \frac{10}{2} + \frac{5-1}{4} = 6$  arestas.

Com respeito ao problema SIGNED MAX CUT, Poljak e Turzík [54] provaram que  $\frac{m}{2} + \frac{n-1}{4}$  é uma cota inferior justa para o tamanho (número de arestas) de um subgrafo balanceado de qualquer grafo  $G$  conexo sinalizado. Devido a esse fato, representaremos essa cota inferior utilizando as iniciais de Poljak e Turzik, isto é,  $pt(G) = \frac{m}{2} + \frac{n-1}{4}$ . Motivados pelo resultado anterior para MAX CUT, Crowston, Gutin, Jones e Muciaccia [15] consideraram a seguinte parametrização de SIGNED MAX CUT, que nos fornece um corte de arestas de tamanho maior ou igual a  $\frac{m}{2} + \frac{n-1}{4}$ , sendo por isso chamada de parametrização *acima do limite inferior justo* a qual utilizamos nesse capítulo.

SIGNED MAX CUT ABOVE TIGHT LOWER BOUND(ATLB)

INSTÂNCIA: Um grafo  $G$  conexo sinalizado e um inteiro positivo  $k$ .

PARÂMETRO:  $k$

QUESTÃO:  $G$  contém um subgrafo balanceado com pelo menos  $\frac{m}{2} + \frac{n-1}{4} + \frac{k}{4}$  arestas?

Por motivo de simplificação, o problema acima será referido ao longo do capítulo como SIGNED MAX CUT ATLB. Crowston, Gutin, Jones e Muciaccia [15] provaram que SIGNED MAX CUT ATLB é também FPT, determinando também um núcleo de tamanho  $O(k^3)$  para grafos em geral. Como já observamos anteriormente, o problema análogo MAX CUT ATLB é obtido a partir de SIGNED MAX CUT ATLB, considerando o grafo sinalizado  $G$  com todas as arestas negativas. Desse modo, os resultados de Crowston, Gutin, Jones e Muciaccia [15] melhoraram consideravelmente o tamanho do núcleo que, antes era de  $O(k^5)$  no caso particular de grafos não sinalizados estudados no problema MAX CUT ATLB dado por Crowston, Jones e Mnich [16], para um núcleo de  $O(k^3)$  no caso geral de SIGNED MAX CUT ATLB.

Um grafo  $G$  é  $(r, \ell)$  se  $V(G)$  pode ser particionado em  $r$  *conjuntos independentes* e  $\ell$  *cliques*. O problema de reconhecimento para grafos  $(r, \ell)$  é NP-completo se, e somente se,  $\max\{r, \ell\} \geq 3$  [9, 32].

Nesse capítulo mostramos que SIGNED MAX CUT ATLB tem núcleo de tamanho  $O(k^2)$  em grafos  $(r, \ell)$ , para todo  $r, \ell > 0$  fixados, utilizando fortemente as técnicas desenvolvidas por Crowston, Gutin, Jones e Muciaccia [15] e explorando a estrutura dos grafos  $(r, \ell)$  para melhorar o tamanho do núcleo de cúbico para quadrático.

A seguir listamos algumas definições e resultados obtidos para grafos em geral e que utilizamos nas próximas sessões.

## 4.1 Resultados Gerais

Dado um grafo simples  $G = (V, E)$ , se  $S \subset V(G)$  definimos  $G - S = G[V(G) \setminus S]$ . Se o grafo  $G$  é *sinalizado*, dizemos que um ciclo  $C$  de  $G$  é *positivo* quando tem um

número *par* de arestas negativas (veja a Figura 4.5).

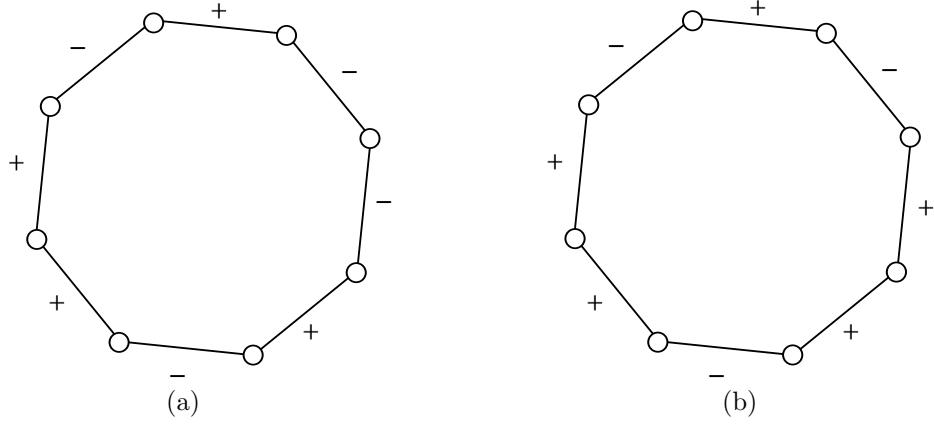


Figura 4.5: (a) Um ciclo positivo e (b) um ciclo negativo.

**Teorema 4.1** (Harary). [46] *Um grafo sinalizado  $G$  é balanceado se, e somente se, todo ciclo de  $G$  é positivo.*

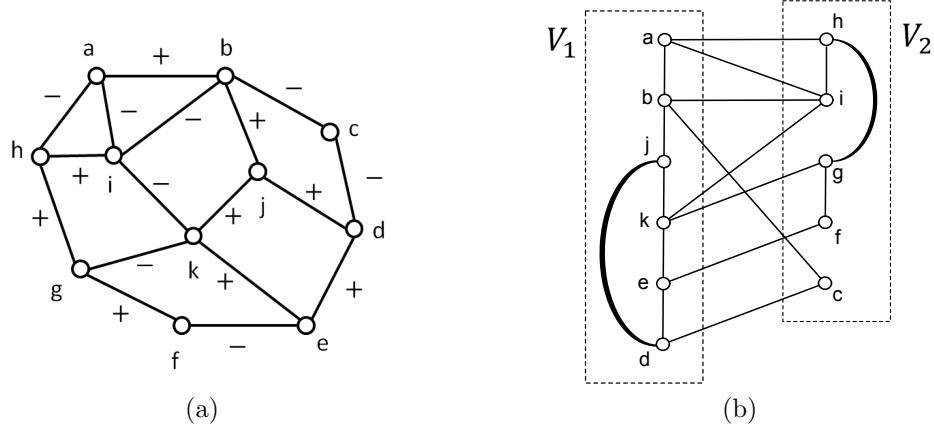


Figura 4.6: (a) Um grafo sinalizado balanceado e (b) sua partição balanceada.

O Lema 4.2 a seguir, devido a Crowston, Gutin, Jones e Muciaccia [15] nos será de grande utilidade, uma vez que fornece uma cota inferior para o tamanho de um subgrafo sinalizado balanceado de um grafo dado. Seja  $G = (V, E)$  um grafo sinalizado. Representaremos por  $\beta(G)$  o tamanho do subgrafo balanceado  $H$  de  $G$  que tem o maior número de arestas  $\beta(G) = |E(H)|$ .

**Lema 4.2.** [15] *Seja  $G = (V, E)$  um grafo conexo sinalizado e  $(U, W)$  uma partição não-trivial de  $V(G)$ . Então  $\beta(G) \geq \beta(G[U]) + \beta(G[W]) + \frac{1}{2}|U, W|$ . Além disso, se  $G[U]$  e  $G[W]$  têm, respectivamente,  $c_1$  e  $c_2$  componentes conexas,  $\beta(G[U]) \geq pt(G[U]) + \frac{k_1}{4}$  e  $\beta(G[W]) \geq pt(G[W]) + \frac{k_2}{4}$ , então  $\beta(G) \geq pt(G) + \frac{k_1+k_2-(c_1+c_2-1)}{4}$ .*

## 4.2 Núcleos

O processo de *kernelização* é uma técnica utilizada para reduzir o tamanho de uma instância  $(G, k)$  para o problema dado por meio de regras de redução de dados em tempo polinomial que transformam a instância original  $(G, k)$  em uma nova instância  $(G', k')$  de tamanho limitado por uma função do parâmetro  $k$ . Para os problemas estudados nesse capítulo, consideraremos o tamanho de  $G'$  como sendo o número de vértices de  $G'$ . A instância reduzida  $(G', k')$  é o tão mencionado *núcleo* ou *kernel* do problema. Uma vez que o *núcleo* do problema tenha sido obtido, sabemos que o problema é tratável por parâmetro fixo, já que o tempo de execução de qualquer algoritmo de força bruta depende exclusivamente do parâmetro  $k$ . A recíproca também é verdadeira: toda vez que um problema parametrizado é FPT, então admite um *núcleo* [20, 34, 51]. A questão crucial com respeito a *núcleos* é decidir quando um problema parametrizado admite um *núcleo polinomial* ou NÃO. Um núcleo é polinomial quando seu tamanho é de ordem polinomial, ou seja, quando existe um inteiro positivo  $p$  tal que o tamanho do núcleo é de  $O(k^p)$ .

Crowston, Gutin, Jones e Muciaccia [15] mostraram que SIGNED MAX CUT ATLB é FPT para grafos em geral através de um algoritmo com tempo de execução  $2^{3k}n^{O(1)}$ . O algoritmo aplica algumas regras de redução à instância  $(G, k)$  que ou determinam se essa instância tem resposta SIM para o problema de decisão SIGNED MAX CUT ATLB, ou produzem um subconjunto  $S$  com no máximo  $3k$  vértices tal que  $G - S$  é um *grafo bloco* ou *floresta de cliques* (um grafo no qual toda componente biconexa é uma clique) sem arestas positivas. Esse resultado está estabelecido a seguir.

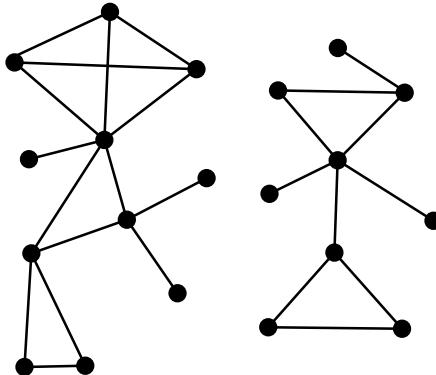


Figura 4.7: Exemplo de um grafo bloco.

**Proposição 4.3.** [15] Se  $(G, k)$  é uma instância de SIGNED MAX CUT ATLB, então é possível determinar em tempo polinomial que  $(G, k)$  é uma instância SIM ou que existe um subconjunto  $S$  com no máximo  $3k$  vértices tal que  $G - S$  é um grafo bloco sem arestas positivas.

Existem dois tipos de regras de redução aplicadas por Crowston, Gutin, Jones e

Muciaccia [15] a uma instância  $(G, k)$  com o objetivo de obter um *núcleo* de tamanho  $O(k^3)$  para SIGNED MAX CUT ATLB: regras de redução do tipo *one-way* e do tipo *two-way*.

Uma regra de redução é do tipo *two-way* se sua aplicação na instância  $(G, k)$  produz, em tempo polinomial no tamanho de  $G$ , uma instância  $(G', k')$  equivalente a  $(G, k)$ , ou seja,  $(G, k)$  é uma instância SIM se, e somente se,  $(G', k')$  é uma instância SIM. Isso significa que essas regras podem certamente ser aplicadas a qualquer instância do problema em questão para a obtenção de algoritmos FPT ou *núcleos*, na medida que o parâmetro  $k$  NÃO aumenta pela aplicação das regras definidas por Crowston, Gutin, Jones e Muciaccia [15].

Por outro lado, uma regra de redução do tipo *one-way*, quando aplicada a uma instância  $(G, k)$ , produz uma instância  $(G', k')$  que satisfaz a seguinte condição: se  $(G', k')$  é uma instância SIM, então  $(G, k)$  é uma instância SIM. Note que é possível que  $(G, k)$  seja uma instância SIM e  $(G', k')$  seja uma instância NÃO. Embora as regras de redução do tipo *one-way* não produzam uma instância equivalente a  $(G, k)$ , seu uso repetido, juntamente com as regras de redução do tipo *two-way* nos permitem *em determinados casos*, encontrar uma instância SIM  $(G', k')$  com  $k' \leq 0$  e assim concluir certamente que  $(G, k)$  é uma instância SIM. A Figura 4.8 ilustra o funcionamento das regras de redução.

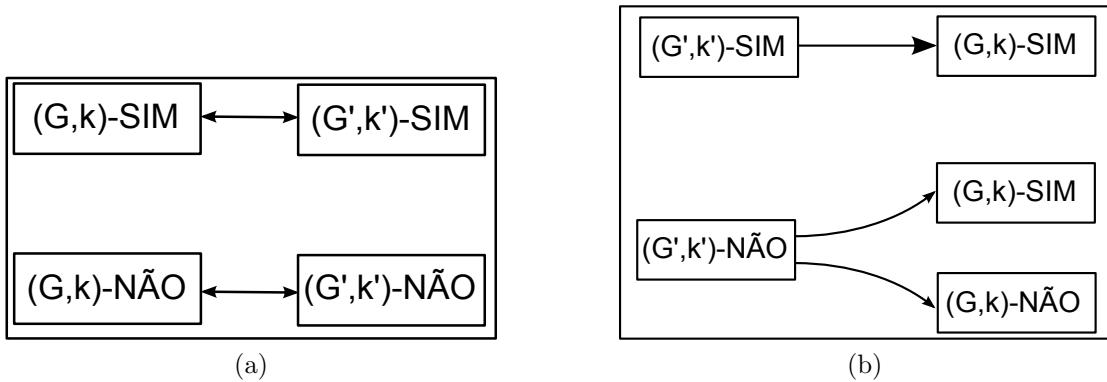


Figura 4.8: (a) Regras de redução *two-way* (b) Regras de redução *one-way*.

A seguir, listamos as primeiras sete dentre as onze regras de redução criadas por Crowston, Gutin, Jones e Muciaccia [15]. Essas regras são do tipo *one-way*, enquanto as regras de 8 a 11 que listamos posteriormente são do tipo *two-way*. O algoritmo construído por Crowston, Gutin, Jones e Muciaccia [15] aplica estas regras de redução ao grafo de entrada do problema parametrizado, marcando alguns vértices nos casos em que o parâmetro  $k$  é decrementado.

**Regra 1.** *Seja  $G$  um grafo conexo sinalizado. Se  $abc$  é um triângulo positivo tal que  $G - \{a, b, c\}$  é conexo, então marque os vértices  $a, b, c$ , remova-os de  $G$  e faça  $k' = k - 3$ .*

**Regra 2.** Seja  $G$  um grafo conexo sinalizado. Se  $abca$  é um triângulo positivo tal que  $G - \{a, b, c\}$  tem duas componentes conexas  $C$  e  $Y$ , sendo  $C$  uma clique que não contém triângulos positivos, então marque os vértices  $a, b, c$ , remova-os de  $G$  juntamente com  $C$  e faça  $k' = k - 2$ .

**Regra 3.** Sejam  $G$  um grafo conexo sinalizado e  $C$  uma clique que que não contém triângulos positivos e é uma componente conexa de  $G - v$  para algum vértice  $v \in G$ . Se existem  $a, b \in V(C)$  tais que  $G - \{a, b\}$  é conexo e  $v$  é adjacente a  $a$ , mas não é adjacente a  $b$ , então marque os vértices  $a, b$ , remova-os de  $G$  e faça  $k' = k - 2$ .

**Regra 4.** Sejam  $G$  um grafo conexo sinalizado e  $C$  uma clique que que não contém triângulos positivos e é uma componente conexa de  $G - v$  para algum vértice  $v \in G$ . Se existem  $a, b \in V(C)$  tais que  $G - \{a, b\}$  é conexo e  $vabv$  é um triângulo positivo, então marque os vértices  $a, b$ , remova-os de  $G$  e faça  $k' = k - 4$ .

**Regra 5.** Sejam  $G$  um grafo conexo sinalizado e  $C$  uma clique que que não contém triângulos positivos e é uma componente conexa de  $G - v$  para algum vértice  $v \in G$ . Se  $G[V(C) \cup \{v\}]$  é uma clique de  $G$  que não contém triângulos positivos, então remova  $C$  de  $G$  e faça  $k' = k$ .

**Regra 6.** Seja  $G$  um grafo conexo sinalizado. Se  $a, b, c \in V(G)$ ,  $\{ab, bc\} \in E(G)$ , mas  $ac \notin E(G)$ , e  $G - \{a, b, c\}$  é conexo, então marque os vértices  $a, b, c$ , remova-os de  $G$  e faça  $k' = k - 1$ .

**Regra 7.** Sejam  $G$  um grafo conexo sinalizado,  $C$  e  $Y$  duas componentes conexas de  $G - \{v, b\}$  para algum par de vértices  $v, b \in V(G)$  tal que  $vb \notin E(G)$  e  $C$  é uma clique que não contém triângulos positivos. Se  $G[V(C) \cup \{v\}]$  e  $G[V(C) \cup \{b\}]$  são cliques que não contêm triângulos positivos, então marque os vértices  $v, b$ , remova-os de  $G$  juntamente com  $C$  e faça  $k' = k - 1$ .

Vamos formular uma nova regra de redução do tipo *one-way* que generaliza a Regra 6 do tipo *one-way* dada por Crowston, Gutin, Jones e Muciaccia [15].

**Regra 6<sup>+</sup>.** Seja  $G$  um grafo conexo. Se  $v \in V(G)$  e  $u_1, u_2, \dots, u_c$  são vizinhos de  $v$  dois a dois NÃO adjacentes tais que  $c \geq 2$  e  $G - \{v, u_1, u_2, \dots, u_c\}$  é conexo, então delete  $v, u_1, u_2, \dots, u_c$  e faça  $k' = k - c + 1$ .

Note que a Regra 6 de Crowston, Gutin, Jones e Muciaccia [15] corresponde exatamente ao caso  $c = 2$  da Regra 6<sup>+</sup> definida acima.

**Definição 4.4.** Uma regra de redução do tipo *one-way* é **segura** se não transforma uma instância NÃO em uma instância SIM.

Crowston, Gutin, Jones e Muciaccia [15] provaram que as regras de redução de 1 a 7 são todas seguras. Nós provamos que a regra 6<sup>+</sup> também é segura.

**Lema 4.5.** *A Regra 6<sup>+</sup> é segura.*

*Demonastração.* Sejam  $v \in V(G)$  e  $u_1, u_2, \dots, u_c$  como na descrição da Regra 6<sup>+</sup>. Consideremos o grafo  $P = G[\{v, u_1, u_2, \dots, u_c\}]$ . Observe que  $pt(P) = \frac{c}{2} + \frac{c}{4} = \frac{3c}{4}$  e, uma vez que  $P$  é uma estrela, temos  $P$  balanceado, quaisquer que sejam os sinais das arestas de  $P$ , de acordo com o Teorema 4.1. Portanto,  $\beta(P) = c = pt(P) + \frac{c}{4}$ . Seja  $G'$  o grafo obtido de  $G$  pela remoção dos vértices  $v, u_1, u_2, \dots, u_c$ . Temos  $G'$  conexo por hipótese. Suponha que  $\beta(G') \geq pt(G') + \frac{k'}{4}$ , em que  $k' = k - c + 1$ . Então pelo Lema 4.2,  $\beta(G) \geq pt(G) + \frac{k'+c-1}{4} = pt(G) + \frac{k}{4}$ , e consequentemente  $G$  é uma instância SIM.  $\square$

Na próxima seção mostramos que SIGNED MAX CUT ATLB admite um *núcleo* de tamanho  $O(k^2)$  em grafos  $(r, \ell)$ , quaisquer que sejam os inteiros não negativos  $r$  e  $\ell$ . Para isso, usamos fortemente os resultados e técnicas desenvolvidos por Crowston, Gutin, Jones e Muciaccia [15].

### 4.3 Núcleo Quadrático para Grafos $(r, \ell)$

O algoritmo de *nucleação* inicialmente aplica à instância  $(G, k)$ , o algoritmo dado na Proposição 4.3. Este algoritmo usa as regras de redução do tipo *two-way*: Regra 8, Regra 9, Regra 10 e Regra 11, definidas por Crowston, Gutin, Jones e Muciaccia [15]. A saber, o algoritmo aplica exaustivamente todas essas regras de redução até que nenhuma delas possa mais ser aplicada. Nesse momento, há duas alternativas para o processo: ou pode-se concluir diretamente que a instância  $(G, k)$  é uma instância SIM (pois o parâmetro final  $k'$  obtido após todas as reduções é menor ou igual a zero), ou, sabendo que o grafo não pode mais ser reduzido, prova-se que seu tamanho pode ser limitado por uma função que depende exclusivamente de  $k$ . É exatamente nesse ponto que exploramos a estrutura dos grafos  $(r, \ell)$  a fim de obter um *núcleo* quadrático para tais grafos. Reforçamos, nesse ponto do raciocínio, que NÃO se faz necessária a obtenção algorítmica de uma tal partição de  $V(G)$  em  $r$  conjuntos independentes e  $\ell$  cliques, uma vez que precisamos apenas garantir a existência de tal partição para efetuarmos a *análise* do algoritmo.

Reproduzimos aqui as definições de duas das quatro regras de redução mencionadas anteriormente, pois serão necessárias nas provas de dois teoremas subsequentes. As demais regras podem ser encontradas em [15]. A seguir introduzimos algumas definições de [15] necessárias para o estabelecimento das regras de redução que utilizaremos.

Para um bloco  $C$  em  $G - S$ , sejam  $C_{\text{int}} = \{x \in V(C) \mid N_{G-S}(x) \subseteq V(C)\}$  o *interior* de  $C$  e  $C_{\text{ext}} = V(C) \setminus C_{\text{int}}$  o *exterior* de  $C$ . Se um bloco  $C$  tem apenas dois vértices e esses vértices pertencem a  $C_{\text{ext}}$ , então  $C$  é chamado *bloco-caminho*. Um

vértice que pertence apenas a *bloco-caminho* é chamado *vértice-de-caminho*. Um bloco  $C$  de  $G - S$  que tem  $|C_{\text{ext}}| \leq 1$  é chamado *bloco-folha* (veja a Figura 4.9). Agora estamos em condições de escrever as regras de redução do tipo *two-way*.

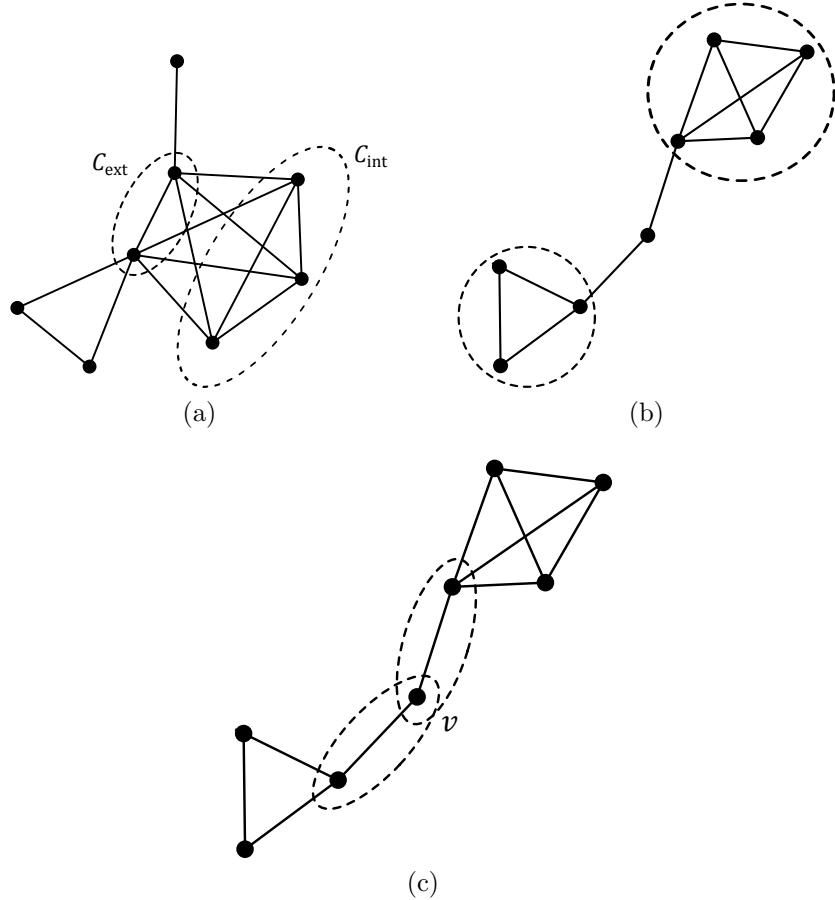


Figura 4.9: (a) Um bloco  $C$  com  $C_{\text{int}}$  e  $C_{\text{ext}}$ , (b) blocos folha de um grafo e (c) blocos caminho e o vértice de caminho  $v$ .

Listamos a seguir as regras de redução de 8 a 11 de Crowston, Gutin, Jones e Muciaccia [15]. Essas são todas do tipo *two-way*.

**Regra 8.** *Seja  $C$  um bloco de  $G - S$ . Se existe  $X \subseteq C_{\text{int}}$  tal que  $|X| > \frac{|V(C)| + |N_G(X) \cap S|}{2} \geq 1$ ,  $N_G^+(x) \cap S = N_G^+(X) \cap S$  e  $N_G^-(x) \cap S = N_G^-(X) \cap S$  para todo  $x \in X$ , então delete dois vértices arbitrários  $x_1, x_2 \in X$  e faça  $k' = k$ .*

**Regra 9.** *Seja  $C$  um bloco de  $G - S$ . Se  $|V(C)|$  é par e existe  $X \subseteq C_{\text{int}}$  tal que  $|X| = \frac{|V(C)|}{2}$  e  $N_G(X) \cap S = \emptyset$ , então delete um vértice  $x \in X$ , e faça  $k' = k - 1$ .*

**Regra 10.** *Seja  $C$  um bloco de  $G - S$  com conjunto de vértices  $\{x, y, u\}$ , tal que  $N_G(u) = \{x, y\}$ . Se a aresta  $\{x, y\}$  é uma ponte em  $G - u$ , delete  $C$ , adicione um novo vértice  $z$ , arestas positivas  $\{\{z, v\} \mid v \in N_{G-u}^+(\{x, y\})\}$ , arestas negativas  $\{\{z, v\} \mid v \in N_{G-u}^-(\{x, y\})\}$ , e faça  $k' = k$ . Caso contrário, delete  $u$  e a aresta  $\{x, y\}$ , e faça  $k' = k - 1$ .*

**Regra 11.** Seja  $T$  uma componente conexa de  $G - S$  que é adjacente apenas ao vértice  $s \in S$ . Forme uma instância de MAX-CUT-WITH-WEIGHTED-VERTICES em  $T$ , definindo  $w_1(x) = 1$  se  $x \in N_G^+(s) \cap T$  ( $w_1(x) = 0$ , caso contrário), e  $w_2(y) = 1$  se  $y \in N_G^-(s) \cap T$  ( $w_2(y) = 0$ , caso contrário). Se  $\beta(G[V(T) \cup \{s\}]) = pt(G[V(T) \cup \{s\}]) + \frac{p}{4}$ , então delete  $T$  e faça  $k' = k - p$ .

Na Regra 10 acima,  $N^+$  (respectivamente  $N^-$ ) representam as vizinhanças dos conjuntos dados via arestas positivas (respectivamente negativas). A Figura 4.10 ilustra a aplicação da Regra 10.

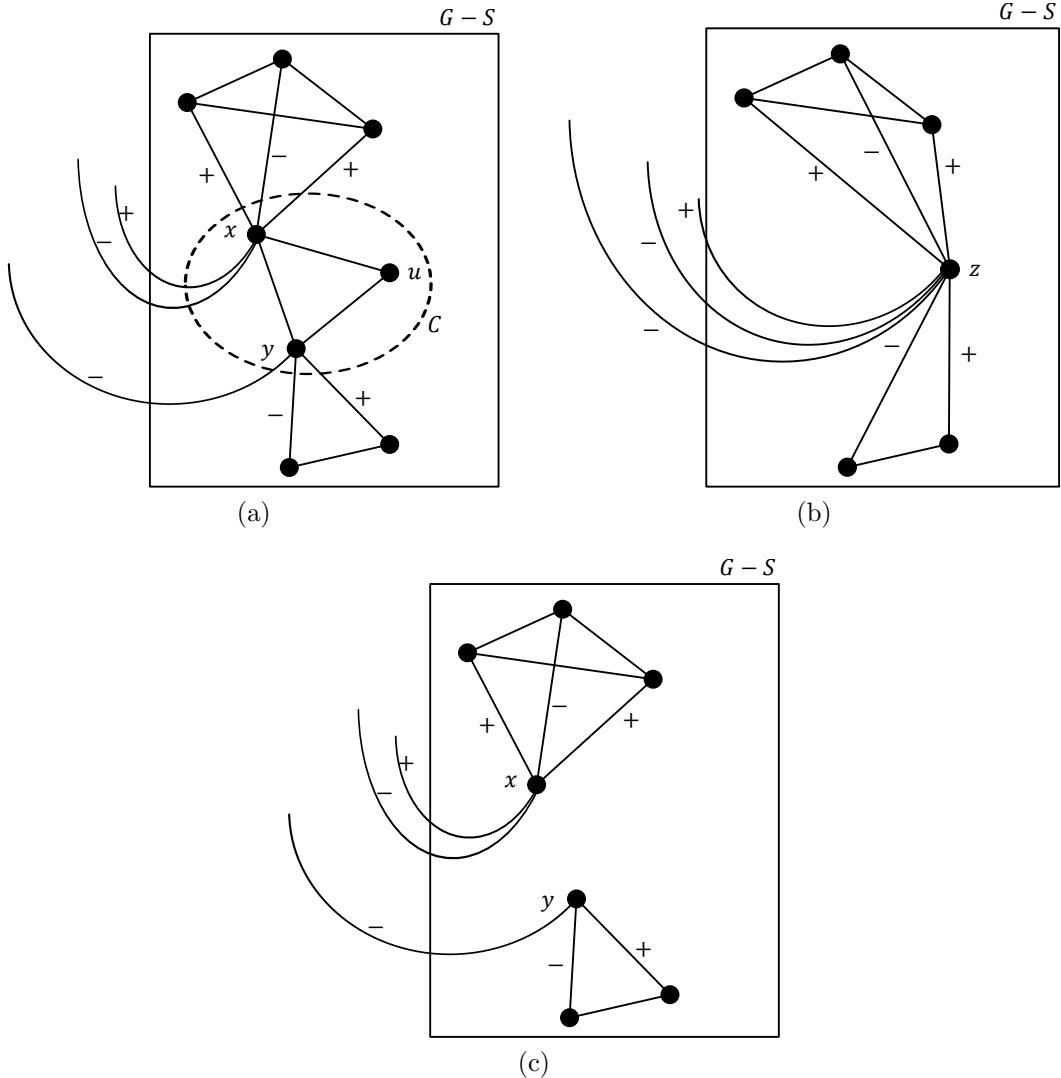


Figura 4.10: (a) Um bloco  $C = \{x, y, u\}$  em  $G - S$  nas condições da Regra 10. Se a aresta  $\{x, y\}$  é uma ponte em  $G - u$ , procedemos como em (b). Caso contrário, fazemos a transformação ilustrada em (c).

**Definição 4.6.** Uma regra de redução do tipo two-way é **válida** se transforma instâncias SIM em instâncias SIM e instâncias NÃO em instâncias NÃO.

Precisamos dos seguintes resultados de Crowston, Gutin, Jones e Muciaccia [15]:

**Lema 4.7.** [15] Seja  $T$  uma componente conexa de  $G - S$ . Então, para cada bloco-folha  $C$  de  $T$ , tem-se  $N_G(C_{int}) \cap S \neq \emptyset$ . Além disso, se  $|N_G(S) \cap V(T)| = 1$ , então  $T$  é formado por um único vértice.

Seja  $\mathcal{B}$  o conjunto dos blocos de  $G - S$  que não são blocos-de-caminho.

**Corolário 4.8.** [15] Se  $\sum_{C \in \mathcal{B}} |N_G(C_{int}) \cap S| \geq |S|(2|S| - 3 + 2k) + 1$ , então a instância  $(G, k)$  é uma instância SIM. Se  $\sum_{C \in \mathcal{B}} |N_G(C_{int}) \cap S| \sqrt{?} < |S|(2|S| - 3 + 2k) + 1$ , então  $\sum_{C \in \mathcal{B}} |N_G(C_{int}) \cap S| \leq 3k(8k - 3)$ .

**Corolário 4.9.** [15]  $G - S$  contém no máximo  $6k(8k - 3)$  blocos que não são blocos-de-caminho e  $24k^2(8k - 3)$  vértices-de-caminho.

**Corolário 4.10.** [15]  $G - S$  contém no máximo  $12k(8k - 3)$  vértices no exterior dos blocos que não são blocos-de-caminho.

**Lema 4.11.** [15] Para um bloco  $C$  se  $|V(C)| \geq 2|C_{ext}| + |N_G(C_{int}) \cap S|(2|S| + 2k + 1)$ , então  $(G, k)$  é uma instância SIM. Se  $|V(C)| < 2|C_{ext}| + |N_G(C_{int}) \cap S|(2|S| + 2k + 1)$ , então  $|V(C)| \leq 2|C_{ext}| + |N_G(C_{int}) \cap S|(8k + 1)$ .

Por motivos de simplificação, apresentamos o processo de obtenção do núcleo quadrático para grafos *split* (isto é, quando  $r = \ell = 1$ ) na Subseção 4.3.1. Em seguida, generalizamos para  $r$  e  $\ell$  arbitrários na Subseção 4.3.2.

### 4.3.1 Núcleo Quadrático em Grafos Split

Mostramos a seguir que o problema SIGNED MAX CUT ATLB tem núcleo quadrático quando considerado em grafos *split*.

**Teorema 4.12.** SIGNED MAX CUT ATLB em grafos *split* admite um núcleo com  $O(k^2)$  vértices.

*Demonstração.* Consideramos como instância  $G = (V, E)$  um grafo *split* e  $k$  um inteiro positivo. Aplicando o algoritmo de tempo polinomial da Proposição 4.3, podemos concluir que ou  $(G, k)$  é uma instância SIM ou, no caso em que o conjunto  $S$  é definido, vamos construir um grafo  $(G', k')$  tal que  $|V(G')| \leq 168k^2 + 24k$  e  $k'$  é polinomialmente limitado por  $k$ , isto é,  $(G', k')$  é o núcleo polinomial procurado e  $G - S$  é uma floresta de cliques ou grafo bloco sem arestas positivas.

Como  $G$  é *split* e ser *split* é uma propriedade hereditária, temos que  $G[S]$  e  $G - S$  são também grafos *split*. Sejam  $S = K^1 \cup I^1$ ,  $V(G) - S = K^2 \cup I^2$ , tais que  $K^1, K^2$  são cliques e  $I^1, I^2$  são conjuntos independentes. Vamos particionar  $I^2$  em três subconjuntos:

1.  $I_0^2$  contendo apenas os vértices isolados de  $G - S$ ;

2.  $I_1^2$  contendo os vértices de grau um em  $G - S$  e
3.  $I_u^2$  contendo no máximo um vértice  $u$  adjacente a  $K^2$  em  $G - S$ , isto é,  $N_{G-S}(u) = K^2$ .

Veja a Figura 4.11 para ilustração.

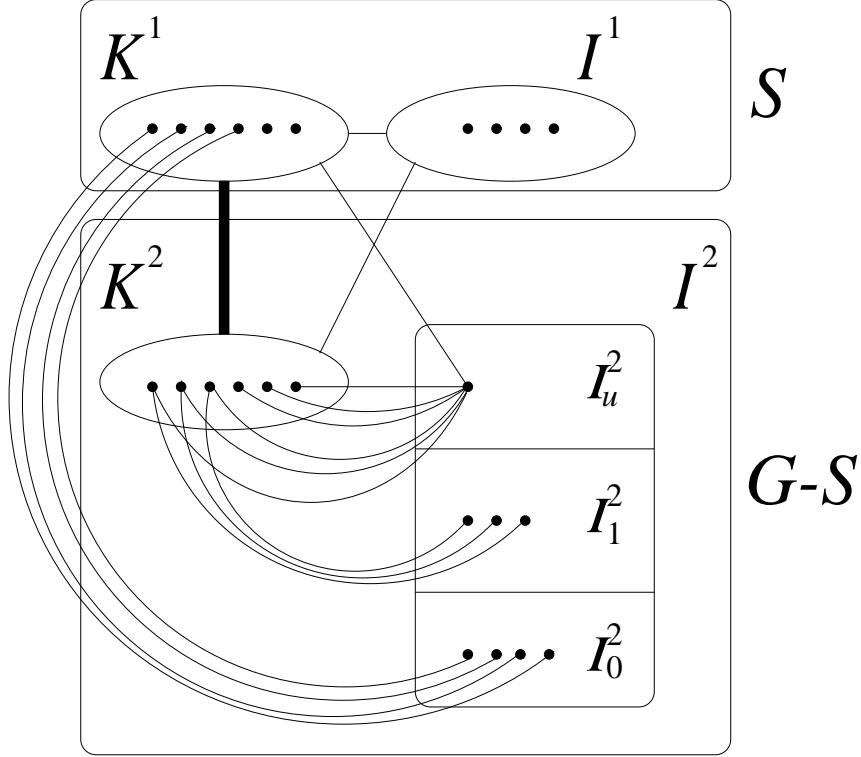


Figura 4.11: Grafo bloco  $G - S$  sem arestas positivas obtido pelo algoritmo parametrizado FPT, mencionado na Proposição 4.3, para o problema do balanceamento devido à Crowston, Gutin, Jones e Muciaccia [15], obtido da entrada  $(G, k)$  na qual  $G$  é um grafo split sinalizado quando o algoritmo não responde expressamente SIM.

De fato, os conjuntos  $I_0^2$ ,  $I_1^2$  e  $I_u^2$  definem uma partição de  $I^2$ , pois não é possível encontrar um vértice  $v \in I^2$  tal que  $1 < d_{K^2}(v) < |V(K^2)|$ , uma vez que isso implicaria em termos um bloco em  $G - S$  que não é uma clique, mas seria um bloco formado por duas cliques ( $K^2$  e  $\{v\} \cup N_{K^2}(v)$ ) que têm como interseção o conjunto  $N_{K^2}(v)$ , o que contrariaria o fato de que  $G - S$  é um grafo bloco. Isso contraria a estrutura do grafo bloco, pois aí, cada bloco é uma clique maximal e dois blocos só podem se intersectar em no máximo um vértice em uma floresta de cliques. Portanto não existem vértices  $v \in I^2$  tal que  $1 < d_{K^2}(v) < |V(K^2)|$ , e assim  $(I_0^2, I_1^2, (I_u^2)) = I^2$  é uma partição para  $I^2$ .

Note que  $G - S$  tem, no máximo, um bloco-caminho. Essa possibilidade só ocorre quando a clique  $K^2$  é exatamente o grafo completo  $K_2$ , o conjunto  $I_u^2$  é vazio e cada um dos vértices de  $K_2$  tem pelo menos um vizinho em  $I_1^2$ . Nesse caso, cada uma das arestas da forma  $\{v, w\}$  com  $v \in K_2$  e  $w \in N_{I^2}(v)$  é um bloco-folha. Portanto, neste

caso, o único bloco-caminho é a clique  $K_2$  e  $G - S$  não tem vértices-de-caminho. A Figura 4.12 ilustra essa situação.

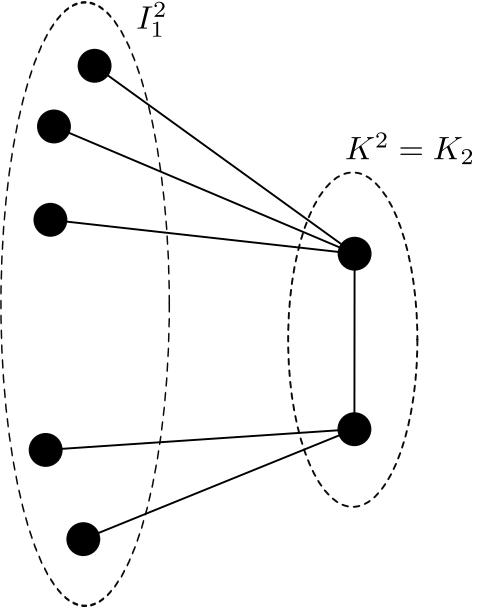


Figura 4.12: Único bloco caminho possível em  $G - S$ , quando  $K^2 = K_2$  e cada um dos vértices de  $K^2$  tem pelo menos um vizinho em  $I_1^2$ .

Para cada vértice  $x$  no conjunto  $I_1^2$ , existe um bloco-folha  $C^x$  tal que  $|V(C^x)| = 2$  e  $C_{\text{int}}^x = \{x\}$ . Desse modo pode-se aplicar a Regra 9 ao grafo  $G - S$ , de modo que para cada vértice em  $I_1^2$  sem vizinhos em  $S$ , deletamos sucessivamente cada um desses vértices e obtemos  $k' = k - |Q|$ , para  $Q = \{x \in I_1^2 | N_S(x) = \emptyset\}$ , dando origem a um novo grafo  $G'$  e atualizando o parâmetro  $k'$ . Note que a diminuição do parâmetro  $k$  em  $|Q|$  unidades significa que cada uma das  $|Q|$  arestas pertencentes aos blocos-folha serão imediatamente alocadas no subgrafo sinalizado balanceado máximo de  $G$ . A Figura 4.13 ilustra essa situação.

Pode-se também deletar diretamente todos os vértices de  $I_0^2$  sem vizinhos em  $S$ , pois estes são vértices isolados no grafo  $G$  e por não serem extremos de nenhuma aresta, não contribuem para o tamanho do subgrafo sinalizado balanceado máximo de  $G$ . Nesse caso, fazemos  $k' = k$  já que não há remoção de nenhuma aresta.

Observe que a Regra 10 só pode ser aplicada em dois casos: primeiro, quando  $K^1 = \emptyset$  e  $K^2$  é um triângulo; segundo, quando  $K^1 = \emptyset$ ,  $K^2 = \{x, y\}$  e  $I_u^2 = \{u\}$ . De fato, como o conjunto  $\{x, y, u\} \subset G - S$  mencionado na Regra 10 forma um triângulo tal que  $N_G(u) = \{x, y\}$ , não podemos ter mais de um vértice desse conjunto compondo  $I^2$ , o que resulta em  $K^2 = \{x, y, u\}$ , (correspondente ao primeiro caso), ou então  $K^2 = \{x, y\}$  e  $I_u^2 = \{u\}$  (correspondente ao segundo caso). Além disso, como  $K^1 \cup K^2 = K$ , obrigatoriamente precisamos que se tenha  $K^1 = \emptyset$ , para que seja satisfeita a condição  $N_G(u) = \{x, y\}$ . No primeiro caso, quando  $K^1 = \emptyset$  e  $K^2$  é um triângulo, se os vértices  $x$  e  $y$  não possuem nenhum vizinho comum em  $I$ , então

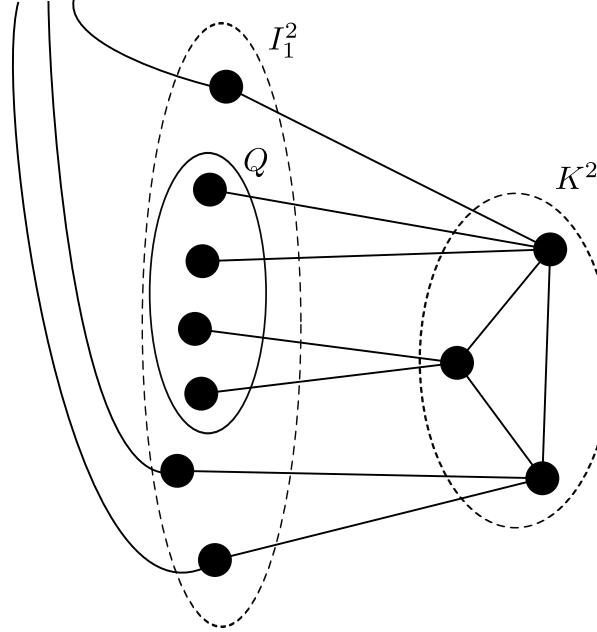


Figura 4.13: Vértices do conjunto  $Q \subset I_1^2$  que não possui vizinhos em  $S$ .

a aresta  $\{x, y\}$  é uma ponte em  $G - u$  e a aplicação da Regra 10 mantém o parâmetro inalterado, sendo  $k' = k$ . Porém, se existir pelo menos um vértice  $v \in V(G)$  que seja adjacente a  $x$  e  $y$ , teremos obrigatoriamente  $v \in S$ , mais ainda,  $v \in I^1$ , pois como já vimos anteriormente  $K_1 = \emptyset$  e não existe vértice em  $I^2$  de grau estritamente maior que 1 e menor que o tamanho de  $K^2$ . Neste caso, a aresta  $\{x, y\}$  não é uma ponte em  $G - u$  e a aplicação da Regra 10 decrementa o parâmetro  $k$  de uma unidade, ou seja,  $k' = k - 1$ .

Usando o Lema 4.7, é possível ver que após a aplicação da Regra 9, cada um dos vértices restantes em  $I_1^2$  é adjacente a algum vértice de  $S$ . Caso contrário, a Regra 9 poderia ser aplicada novamente e deletaríamos os vértices  $x \in C_{\text{int}}^x$  de  $I_1^2$  com  $N_G(C_{\text{int}}^x) \cap S = \emptyset$ , contrariando a hipótese de que o grafo  $G$  já foi reduzido ao máximo pelas regras de redução do tipo *two-way*.

Seja  $\mathcal{B}$  o conjunto dos blocos de  $G - S$  que não são blocos-de-caminho. Como já foi dito anteriormente e ilustrado na Figura 4.12, em um grafo split  $G$  só há uma possibilidade de existência de bloco-caminho. Desse modo podemos afirmar que todos os blocos de  $G - S$ , exceto possivelmente um, não são blocos-caminho. Como sabemos que apenas no caso  $K^2 = K_2$ , pode ocorrer um único bloco-caminho, então o número de blocos-caminho é no máximo um, ocorrendo numa situação muito especial. Logo podemos assumir daqui por diante que em geral  $G - S$  não tem nenhum bloco-caminho.

Pelo Corolário 4.8, ou podemos determinar que  $G$  é uma instância SIM ou então  $\sum_{C \in \mathcal{B}} |N_G(C_{\text{int}}) \cap S| \leq 3k(8k - 3)$ . Usando o Lema 4.11 podemos concluir que para

todo bloco  $C$  de  $G - S$ , vale

$$|V(C)| \leq 2|C_{\text{ext}}| + |N_G(C_{\text{int}}) \cap S| \cdot (8k + 1).$$

Então  $|V(G)| \leq |S| + \sum_{C \in \mathcal{B}} |V(C)|$  uma vez que podemos assumir a inexistência de blocos-caminho em  $G - S$ . Como  $G$  é um grafo split,  $G - S$  também o é e nossa análise anterior mostra que  $\mathcal{B}$  é composto por um grande bloco  $C_0$  definido por  $K^2$ , blocos-folha que são arestas com pelo menos um vizinho em  $S$ , e possivelmente vértices isolados. Seja  $\mathcal{B}_*$  o conjunto de blocos-folha de  $\mathcal{B}$ . Podemos reescrever a desigualdade anterior como

$$|V(G)| \leq |S| + |V(C_0)| + \sum_{C \in \mathcal{B}_*} |V(C)|.$$

Em particular, para o grande bloco  $C_0$  temos:

$$|V(C_0)| \leq 2|C_{0\text{ext}}| + |N_G(C_{0\text{int}}) \cap S|(8k + 1) \leq 2|C_{0\text{ext}}| + |S|(8k + 1).$$

E para todo bloco-folha  $C \in \mathcal{B}_*$  temos  $|V(C)| \leq 2$ . Então

$$\sum_{C \in \mathcal{B}_*} |V(C)| \leq 2|\mathcal{B}_*| \leq 2 \sum_{C \in \mathcal{B}_*} |N_G(C_{\text{int}}) \cap S| \leq 2 \cdot 3k(8k - 3)$$

como uma consequência do Corolário 4.8, uma vez que todo bloco  $C \in \mathcal{B}_*$  tem um vértice  $v \in C_{\text{int}}$  tal que  $N_S(v) \neq \emptyset$ . Finalmente, o número de vértices de  $G$  pode ser limitado como segue:

$$\begin{aligned} |V(G)| &\leq |S| + |S|(8k + 1) + 6k(8k - 3) + 2|C_{0\text{ext}}| \\ &\leq 3k + 3k(8k + 1) + 6k(8k - 3) + 2 \sum_{C \in \mathcal{B}} |C_{\text{ext}}| \\ &\leq 3k + 3k(8k + 1) + 6k(8k - 3) + 12k(8k - 3) \\ &= 168k^2 - 24k = O(k^2), \end{aligned}$$

em que a última desigualdade se deve ao Corolário 4.10, o que mostra a existência de um núcleo quadrático para grafos split.  $\square$

### 4.3.2 Generalização para Grafos $(r, \ell)$

Nesta subseção mostramos como as ideias contidas na subseção anterior podem ser generalizadas para grafos  $(r, \ell)$  arbitrários. Seguimos a mesma estratégia que foi usada na demonstração do Teorema 4.12, porém precisaremos de mais alguns argumentos, a maior parte deles para lidar com os vértices-de-caminho em  $G - S$ .

**Teorema 4.13.** *Para quaisquer dois inteiros  $r, \ell \geq 0$ , SIGNED MAX CUT ATLB em grafos  $(r, \ell)$  admite um núcleo com  $O(k^2)$  vértices.*

*Demonstração.* Seja  $G$  um grafo  $(r, \ell)$  para dois inteiros  $r, \ell \geq 0$  dados. Representemos por  $K^1, K^2, \dots, K^\ell$  as cliques e por  $I^1, I^2, \dots, I^r$  os conjuntos independentes de  $G$ . Reforçamos novamente o fato de que a partição de  $G$  será usada apenas para a análise, não havendo a necessidade de conhecer exatamente tal partição. Pela Proposição 4.3 determinamos em tempo polinomial que  $(G, k)$  é uma instância SIM ou, podemos encontrar um conjunto  $S$  com no máximo  $3k$  vértices tal que o subgrafo  $G - S$  é um grafo-bloco sem arestas positivas. Para um grafo  $(r, \ell)$  temos:  $S = K_1^1 \cup \dots \cup K_1^\ell \cup I_1^1 \cup \dots \cup I_1^r$  e  $G - S = K_2^1 \cup \dots \cup K_2^\ell \cup I_2^1 \cup \dots \cup I_2^r$ , mas agora, em contraste com grafos split, podemos ter vértices-de-caminho em  $G - S$ .

Com o objetivo de limitar o tamanho de  $V(G)$ , limitaremos separadamente o tamanho total dos blocos que não são blocos-caminhos e o número de vértices-de-caminho (ou seja, os únicos vértices que não pertencem a blocos que não são blocos-caminho) em  $G - S$ . Representemos novamente por  $\mathcal{B}$  o conjunto dos blocos que não são blocos-caminho, por  $\mathcal{B}_b$  o subconjunto de  $\mathcal{B}$  formado por blocos contendo pelo menos dois vértices do conjunto  $K^1 \cup K^2 \cup \dots \cup K^\ell$ , e  $\mathcal{B}_s = \mathcal{B} \setminus \mathcal{B}_b$ . A estrutura do grafo-bloco implica em  $|\mathcal{B}_b| \leq \ell$ , e em virtude do Lema 4.11 para cada bloco  $C \in \mathcal{B}_b$  ou podemos concluir que  $(G, k)$  é uma instância SIM ou temos

$$|V(C)| \leq 2|C_{\text{ext}}| + |N_G(C_{\text{int}}) \cap S| \cdot (8k + 1) \leq 2|C_{\text{ext}}| + |S|(8k + 1).$$

Então podemos escrever

$$\begin{aligned} \sum_{C \in \mathcal{B}_b} |V(C)| &\leq 2 \sum_{C \in \mathcal{B}_b} |C_{\text{ext}}| + \sum_{C \in \mathcal{B}_b} |S|(8k + 1) \\ &\leq 2 \sum_{C \in \mathcal{B}_b} |C_{\text{ext}}| + \ell|S|(8k + 1) \\ &\leq 24k(8k - 3) + 3\ell k(8k + 1), \end{aligned}$$

em que a última desigualdade segue do Corolário 4.10.

Para cada bloco  $C \in \mathcal{B}_s$  temos  $|V(C)| \leq r + 1$ , já que  $C$  contém apenas vértices dos  $r$  conjuntos independentes  $I^1, I^2, \dots, I^r$  e possivelmente mais um vértice de alguma das cliques. De fato, caso contrário se  $|V(C)| > r + 1$  então existem dois vértices adjacentes no mesmo conjunto independente, o que é impossível. Consequentemente, pelo Corolário 4.9, podemos limitar a soma do número de vértices de  $C \in \mathcal{B}_s$  por

$$\sum_{C \in \mathcal{B}_s} |V(C)| \leq (r + 1) \cdot 6k(8k - 3).$$

Considerando a função  $g(k, r, \ell) = (2r + \ell + 10)3k(8k + 1)$  e utilizando uma

combinação dos limites encontrados nos dois parágrafos anteriores, concluímos que

$$\sum_{C \in \mathcal{B}} |V(C)| = \sum_{C \in \mathcal{B}_b} |V(C)| + \sum_{C \in \mathcal{B}_s} |V(C)| \leq (2r + \ell + 10)3k(8k + 1) = g(k, r, \ell). \quad (4.1)$$

Resta apenas limitar o número de vértices-de-caminho presentes nos blocos de  $G - S$ . Note que não podemos usar diretamente o limite sobre o número de vértices-de-caminho de  $G - S$  dado pelo Corolário 4.9, já que esse limite tem tamanho  $O(k^3)$ , o que não ajuda em nosso objetivo de determinar o *núcleo* quadrático.

Seja  $\mathcal{P}$  o conjunto formado por todos os vértices-de-caminho em  $G - S$ . Com o objetivo de limitar o tamanho de  $\mathcal{P}$ , argumentamos separadamente, primeiro sobre as componentes em  $G[\mathcal{P}]$  que possuem no máximo dois vértices (Afirmação 4.14) e depois sobre o resto (Afirmação 4.15), sendo  $G[\mathcal{P}]$  o subgrafo de  $G$  induzido por  $\mathcal{P}$ .

Um *vértice marcado* é um vértice-de-caminho cuja vizinhança é formada por vértices que não são de caminho, e uma *aresta marcada* é uma aresta cuja contração dá origem a um vértice marcado. Mostramos na Afirmação 4.14 que o número de vértices-de-caminho marcados e o número de arestas-de-caminho marcadas de  $G[\mathcal{P}]$  são ambos limitados pelo número de blocos que não são de caminho, o qual é no máximo  $6k(8k - 3)$  pelo Lema 4.9.

**Afirmação 4.14.** *O número de vértices marcados em  $G[\mathcal{P}]$  e o número de arestas marcadas em  $G[\mathcal{P}]$  são limitados superiormente pelo número de blocos que não são de caminho.*

Sejam  $\mathcal{P}_1$  o subconjunto de  $\mathcal{P}$  formado pelos vértices não-marcados,  $p = |\mathcal{P}_1|$  e  $e_p$  o número de arestas em  $G[\mathcal{P}_1]$ . Note que o grau mínimo dos vértices de  $G[\mathcal{P}_1]$ , que representaremos por  $\delta(G[\mathcal{P}_1])$ , é tal que  $\delta(G[\mathcal{P}_1]) \geq 1$ . Então o número de arestas  $e_p$  de  $G[\mathcal{P}_1]$  é pelo menos  $\frac{p}{2}$  e este valor é atingido quando  $G[\mathcal{P}_1]$  é formado por arestas disjuntas. Precisamos melhorar esse limite de  $\frac{p}{2}$  para que sejamos capazes de formular a Equação (4.2) a seguir, e podemos fazer isso facilmente proibindo esse último caso.

**Afirmação 4.15.** *Se cada componente conexa de  $G[\mathcal{P}_1]$  tem pelo menos três vértices, então  $e_p \geq \frac{2p}{3}$ .*

O subgrafo induzido  $G[\mathcal{P}_1]$  é uma floresta e, consequentemente, pelo Teorema 4.1  $G[\mathcal{P}_1]$  é balanceado. Então  $\beta(G[\mathcal{P}_1]) = e_p$  e  $pt(G[\mathcal{P}_1]) = \frac{e_p}{2} + \frac{p-1}{4}$  supondo que  $G[\mathcal{P}_1]$  é conexo. A Afirmação 4.15 implica que

$$\beta(G[\mathcal{P}_1]) - pt(G[\mathcal{P}_1]) = e_p - \frac{e_p}{2} - \frac{p-1}{4} = \frac{2e_p - p + 1}{4} \geq \frac{\frac{4p}{3} - p + 1}{4} \geq \frac{\frac{p}{3}}{4}. \quad (4.2)$$

**Afirmação 4.16.** *Seja  $(G, k)$  uma instância de SIGNED MAX CUT ATLB. Se  $\mathcal{P}$  é o conjunto dos vértices-de-caminho em  $G - S$ , então o número de componentes*

conexas de  $G[\mathcal{P}]$  é no máximo  $6k(8k - 3)$ . Se  $\mathcal{W} = G - \mathcal{P}$ , então o número de componentes conexas de  $\mathcal{W}$  é no máximo  $6k(8k - 2)$ .

Agora estamos prontos para juntar todos os resultados e concluir a prova do teorema 4.13.

Devido à Afirmação 4.14, acrescentando uma parcela de  $18k(8k - 3)$  ao tamanho do *núcleo* podemos assumir que cada componente de  $G[\mathcal{P}]$  tem pelo menos três vértices, e assim a hipótese da Afirmação 4.15 é válida. Aplicamos o Lema 4.2 ao grafo  $G$  com  $U = \mathcal{P}$ . Temos  $c_1$  componentes conexas de  $G[\mathcal{P}]$ ,  $c_2$  componentes conexas de  $\mathcal{W}$ , e  $\beta(G) \geq pt(G) + \frac{k_1+k_2-(c_1+c_2-1)}{4}$ , em que  $\beta(G[\mathcal{P}]) \geq pt(G[\mathcal{P}]) + \frac{k_1}{4}$  e  $\beta(\mathcal{W}) \geq pt(\mathcal{W}) + \frac{k_2}{4}$ . Claramente vale  $k_2 \geq 0$ . Pela Equação (4.2) temos  $k_1 \geq \frac{p}{3}$  para  $p = |\mathcal{P}_1|$  e, pela Afirmação 4.16, segue que  $c_1 + c_2 \leq 12k(8k - 2) \leq 96k^2$ . Então  $\beta(G) \geq pt(G) + \frac{\frac{p}{3}-192k^2+1}{4}$ . Consequentemente, se  $\frac{p}{3} - 192k^2 + 1 \geq k$ , então  $(G, k)$  é uma instância SIM. Caso contrário,  $\frac{p}{3} \leq k - 1 + 192k^2$ , ou seja, podemos assumir que  $p \leq 576k^2 + 3k - 3$ .

Desse modo, combinando a Equação (4.1) com os resultados obtidos anteriormente, podemos concluir que

$$|V(G)| \leq |S| + \sum_{C \in \mathcal{B}} |V(C)| + |\mathcal{P}| \leq 3k + g(k, r, \ell) + 18k(8k - 3) + p = O(k^2).$$

Finalmente, note que ao longo de toda a demonstração, quando usamos algum resultado de Crowston, Gutin, Jones e Muciaccia [15] no qual uma das possíveis respostas é que  $(G, k)$  seja uma instância SIM, a condição a ser checada se refere somente aos blocos de  $G - S$ , os quais podemos obter em tempo polinomial pela Proposição 4.3. Portanto, como havíamos afirmado, não precisamos computar nenhuma partição de  $V(G)$  em cliques  $K^1, K^2, \dots, K^\ell$  e conjuntos independentes  $I^1, I^2, \dots, I^r$ . Isso conclui a demonstração do teorema.  $\square$

A seguir apresentamos as demonstrações dos lemas embutidos na demonstração do teorema anterior.

**Demonstração da Afirmação 4.14:** Primeiramente vamos lidar com o número de vértices marcados de  $G[\mathcal{P}]$ . Mostramos essa propriedade por indução no número de vértices marcados de  $G[\mathcal{P}]$ . Um vértice é marcado se e somente se é exclusivamente adjacente a vértices que não são de caminho. Contraímos cada bloco que não é de caminho em um vértice preto e representamos cada vértice-de-caminho por um vértice branco. Desse modo, a floresta de cliques de  $G - S$  será representada por uma floresta simples colorida por duas cores: preto e branco. Seja  $p_0$  o número de vértices marcados de  $G[\mathcal{P}]$ . Se  $p_0 = 1$ , então existem pelo menos dois vértices pretos adjacentes ao único vértice branco de  $G[\mathcal{P}]$ . Logo a propriedade é válida e a base da indução está provada. Suponha que a propriedade seja válida quando

$p_0 = j$  para todo  $1 \leq j \leq p$ . Tome  $G[\mathcal{P}]$  tal que o número de vértices marcados seja  $p + 1$  e escolha um desses vértices marcados. Sem perda de generalidade, podemos supor que  $G[\mathcal{P}]$  é conexo. Caso contrário poderíamos trabalhar com cada uma das componentes conexas de  $G[\mathcal{P}]$  separadamente para obter o mesmo resultado. A remoção desse vértice branco desconecta a árvore, produzindo componentes conexas de tamanho menor contendo no máximo  $p$  vértices marcados. Logo, para cada uma dessas componentes conexas a propriedade é válida pela hipótese de indução, ou seja, o número de vértices marcados em cada componente conexa é no máximo igual ao número de blocos que não são de caminho presentes nessa componente menos um. Portanto, o número de vértices marcados presentes no grafo original é limitado superiormente pela soma do números de blocos que não são de caminho menos o número de componentes conexas obtidas, mais um (o vértice removido) e essa cota superior é menor ou igual ao número de blocos que não são de caminho menos um.

Para determinar o número de arestas marcadas de  $G[\mathcal{P}]$ , procedemos de modo semelhante ao descrito acima, contraindo cada bloco que não é de caminho em um vértice preto e representando cada vértice-de-caminho como um vértice branco. Uma aresta marcada será representada por um vértice vermelho, pois de acordo com a definição, sua contração é um vértice marcado. Devido ao fato de que as arestas marcadas funcionam como vértices marcados, os vértices vermelhos possuem como vizinhos apenas vértices pretos. Então, procedendo por indução, como foi feito para os vértices marcados, podemos mostrar que o número de arestas marcadas é limitado superiormente pelo número de blocos que não são de caminho.

**Demonstração da Afirmação 4.15:** Sem perda de generalidade, podemos supor que  $G[\mathcal{P}_1]$  é conexo pelo mesmo motivo exposto na demonstração da Afirmação 4.14. Seja  $p = |\mathcal{P}_1|$ . Como  $G[\mathcal{P}_1]$  é uma árvore, temos  $|E(G[\mathcal{P}_1])| = p - 1$ . Se  $p \geq 3$  então  $\frac{p}{3} \geq 1$ , ou seja,  $p - \frac{2p}{3} \geq 1$  e  $p - 1 \geq \frac{2p}{3}$  como queríamos.

**Demonstração da Afirmação 4.16:** As mesmas ideias usadas na demonstração da Afirmação 4.14 facilmente implicam que o número de componentes conexas de  $G[\mathcal{P}]$  é no máximo  $6k(8k - 3)$ . Uma vez que  $\mathcal{W} \cap (G - S)$  é o complemento de  $G[\mathcal{P}]$  em  $G - S$ , cada vértice de  $\mathcal{W} \cap (G - S)$  pertence a um bloco que não é de caminho. E como o tamanho de  $S$  é no máximo  $3k$ , usando o Corolário 4.9 podemos concluir que o número de componentes conexas de  $\mathcal{W}$  é no máximo  $3k + 6k(8k - 3) \leq 6k(8k - 2)$ .

## 4.4 Núcleo Linear em Subclasses de Grafos Split

Com o objetivo de determinar um *núcleo* de tamanho menor do que o quadrático encontrado em subclasses de grafos  $(r, \ell)$ , apresentamos nesta seção a subclasse dos

grafos  $d^*$ -split para todo inteiro  $d \geq 1$  que possui um *núcleo* linear. Um grafo  $G = (V, E)$  é  $d^*$ -split se  $V$  pode ser particionado em uma clique  $K$  e um conjunto independente  $I$  tal que todo vértice de  $K$  tem pelo menos um vizinho em  $I$ , e todo vértice de  $I$  tem grau no máximo  $d$ .

Demonstramos na Subseção 4.4.1 que o problema permanece NP-completo quando considerado apenas nessa classe de grafos para todo  $d \geq 2$  (de fato, provamos que mesmo MAX CUT é NP-completo), e na Subseção 4.4.2 exibimos o algoritmo de obtenção do núcleo linear para grafos  $d^*$ -split.

#### 4.4.1 Max Cut é NP-completo em Grafos $d^*$ -Split

Nessa subseção provamos que MAX CUT, caso particular de SIGNED MAX CUT, é NP-completo em grafos  $d^*$ -split para todo  $d \geq 2$  fixado. Esse resultado é justo em termos de  $d$ . De fato, para  $d = 0$ , pela definição anterior não existem grafos  $0^*$ -split, e para  $d = 1$ , MAX CUT em grafos  $1^*$ -split é solúvel em tempo polinomial, pois de acordo com a definição de grafos  $1^*$ -split, tais grafos possuem uma estrutura muito precisa. A saber, são formados por uma clique  $K$  arbitrariamente grande na qual todo vértice  $v \in K$  tem um vizinho exclusivo  $v' \in I$  (isto é, um vértice de grau um). Então, uma solução ótima de MAX CUT consiste de uma partição  $(A, B)$  da clique tal que  $|A| = \lfloor \frac{|K|}{2} \rfloor$ ,  $|B| = \lceil \frac{|K|}{2} \rceil$ , e para cada  $v \in K$ , colocamos seu vizinho exclusivo  $v'$  na parte oposta à qual  $v$  pertence.

Provamos que MAX CUT é NP-completo em grafos sem vértices universais. Em seguida, usando a demonstração feita por Bodlaender e Jansen [6] sobre a complexidade de MAX CUT, provamos que este problema é também NP-completo em grafos  $2^*$ -split e, consequentemente, NP-completo em grafos  $d^*$ -split para todo  $d \geq 2$  fixado.

**Lema 4.17.** MAX CUT é NP-completo em grafos sem vértices universais.

*Demonstração.* Obviamente, MAX CUT restrito a grafos sem vértices universais está em NP pois podemos checar em tempo linear o tamanho de um corte de um grafo em geral. Usamos o fato já bem conhecido de que MAX CUT é NP-completo para grafos em geral [40] para mostrar que esse problema é ainda NP-completo para grafos sem vértices universais. Para isso, consideramos um grafo  $(G, k)$  como uma instância de MAX CUT, e construímos um novo grafo  $(G', k') = (2G, 2k)$  em que  $2G$  é formado por duas cópias disjuntas de  $G$ . Obviamente  $G'$  não tem vértices universais por ser desconexo. Sejam  $mc(G)$  e  $mc(G')$  os tamanhos dos cortes máximos de  $G$  e  $G'$ , respectivamente. Vamos mostrar que  $mc(G) \geq k$  se e somente se  $mc(G') \geq 2k$ . A ida é simples pois se  $V(G) = [V_1, V_2]$  é uma partição de  $V(G)$  tal que  $|[V_1, V_2]| \geq k$ , então a mesma partição em cada cópia reproduz uma partição para  $V(G')$  maior ou igual a  $2k$ . Por outro lado se  $mc(G') \geq 2k$ , afirmamos que  $mc(G') = 2 \cdot mc(G)$ . De fato,  $mc(G') \geq 2 \cdot mc(G)$  porque se  $[V_1, V_2]$  é um corte de  $G$  com tamanho máximo,

então  $2 \cdot mc(G)$  é o tamanho do corte  $[2V_1, 2V_2]$  correspondente produzido em  $G'$ ; veja a Figura 4.14 para uma ilustração. Por outro lado, se  $mc(G') > 2 \cdot mc(G)$ , como  $G'$  é formado por duas cópias disjuntas de  $G$ , necessariamente uma das cópias de  $G$  possui um corte de tamanho maior que  $mc(G)$ , uma contradição.  $\square$

A construção dada na demonstração do Teorema 4.18 apresentado a seguir é exatamente a mesma feita por Bodlaender e Jansen em [6], mas vamos reproduzi-la aqui pois mostramos que partindo de um grafo sem vértices universais (podemos fazê-lo graças ao Lema 4.17), a redução apresentada fornece um grafo  $2^*$ -split.

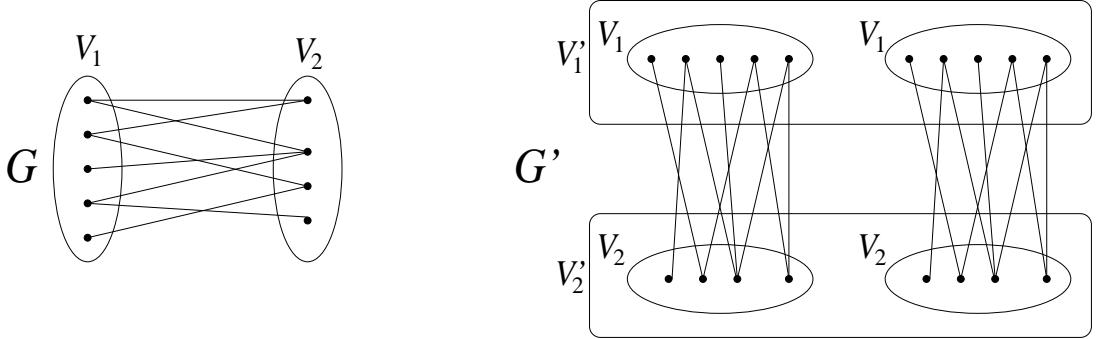


Figura 4.14: Redução para mostrar que MAX CUT é NP-difícil em grafos sem vértices universais.

**Teorema 4.18.** MAX CUT é NP-completo em grafos  $2^*$ -split.

*Demonstração.* Sejam  $G = (V, E)$  um grafo dado e  $\bar{G} = (V, \bar{E})$  o grafo complementar de  $G$ . Seja  $H = (V \cup \bar{E}, F)$  o grafo no qual  $F = \{(v, w) \mid v, w \in V, v \neq w\} \cup \{(v, e) \mid v \in V, e \in \bar{E}, v \text{ é um dos extremos da aresta } e\}$ . Então  $V$  forma uma clique,  $\bar{E}$  forma um conjunto independente em  $H$ , e cada vértice representante da aresta  $e$  é adjacente aos vértices que representam seus extremos. Portanto  $H$  é um grafo split no qual todos os vértices do conjunto independente têm grau dois. Afirmamos que  $G$  admite uma partição com pelo menos  $p$  arestas de corte se, e somente se,  $H$  admite uma partição com pelo menos  $2|\bar{E}| + p$  arestas de corte.

Suponhamos primeiramente que existe uma partição  $(W_1, W_2)$  de  $G$  com pelo menos  $p$  arestas de corte. Particionamos os vértices de  $H$  do seguinte modo: particione  $V$  respeitando a partição original do grafo  $G$ ; para cada  $e \in \bar{E}$ , se ambos os extremos de  $e$  pertencem a  $W_1$ , então ponha  $e$  em  $W_2$ , caso contrário ponha  $e$  em  $W_1$ . É fácil ver que essa partição apresenta o número de arestas de corte desejado.

Agora suponha que temos uma partição  $(W_1, W_2)$  de  $H$  com pelo menos  $2|\bar{E}| + p$  arestas de corte. Particione os vértices de  $G$  em dois subconjuntos  $W_1 \cap V, W_2 \cap V$ .

Essa partição tem o número de arestas de corte desejado. Isso pode ser provado do seguinte modo: para cada aresta  $\{v, w\} \in E$ , temos uma aresta de corte em  $H$  se  $\{v, w\}$  é uma aresta de corte de  $G$ , caso contrário não temos arestas de corte. Para cada aresta  $e = \{v, w\} \in \overline{E}$ , temos que dentre as três arestas  $\{v, w\}, \{e, v\}, \{e, w\}$ , exatamente duas serão arestas de corte. Portanto, o número total de arestas de corte em  $H$  é igual ao número de arestas de corte de  $G$  mais  $2|\overline{E}|$ . Note que  $H$  claramente pode ser construído a partir de  $G$  em tempo polinomial.

Pelo Lema 4.17, podemos assumir que o grafo  $G = (V, E)$  não tem vértices universais. Então o grafo  $\overline{G}$  construído acima não tem vértices isolados, e consequentemente cada vértice  $v \in V$  tem pelo menos um vizinho em  $\overline{E}$ . Uma vez que todo vértice de  $\overline{E}$  tem grau dois, o grafo  $H$  construído na redução anterior é certamente um grafo  $2^*$ -split, o que conclui o teorema.  $\square$

Uma vez que para cada  $d \geq 2$ , a classe dos grafos  $d^*$ -split contém a classe dos grafos  $2^*$ -split, o próximo corolário é uma consequência direta do Teorema 4.18.

**Corolário 4.19.** MAX CUT é NP-completo em grafos  $d^*$ -split para cada  $d \geq 2$  fixo.

#### 4.4.2 O Algoritmo de Kernelização

A seguir, apresentamos na demonstração do Teorema 4.20, um algoritmo de kernelização cujo objetivo é simplesmente a análise do tamanho do núcleo associado ao problema, ou seja, o algoritmo não produz o núcleo.

**Teorema 4.20.** Para qualquer inteiro fixo  $d \geq 1$ , SIGNED MAX CUT ATLB em grafos  $d^*$ -split admite um núcleo com no máximo  $4(d+1)k$  vértices.

*Demonstração.* Dado um grafo sinalizado  $d^*$ -split  $G = (V, E)$  e um inteiro positivo  $k$ , seja  $(K, I)$  uma partição de  $V$  em uma clique  $K$  e um conjunto independente  $I$  tal que cada vértice de  $I$  tem grau no máximo  $d$  e cada vértice de  $K$  tem pelo menos um vizinho em  $I$ . Note que não precisamos exibir tal partição  $(K, I)$  de  $V$ , já que vamos usá-la apenas para a análise.

Sejam  $K_h$  o conjunto dos vértices de  $K$  que têm pelo menos dois vizinhos em  $I$ ,  $K_s = K \setminus K_h$  (por hipótese, cada vértice de  $K_s$  tem exatamente um vizinho em  $I$ ),  $I_h = N_I(K_h)$ , e  $I_s = I \setminus I_h$ . Como assumimos que  $G$  é conexo, valem as desigualdades  $|I_s| \leq |K_s| \leq |K|$ . Agora faremos duas afirmações que nos permitirão certificar em alguns casos que  $(G, k)$  é uma instância SIM. As provas dessas afirmações serão dadas logo após a demonstração deste teorema.

**Afirmiação 4.21.** Se  $|K| \geq (d+1)k$ , então  $(G, k)$  é uma instância SIM.

**Afirmiação 4.22.** Se  $|I_h| \geq 2dk$ , então  $(G, k)$  é uma instância SIM.

Desse modo, se ocorrer pelo menos uma das hipóteses dadas nas afirmações anteriores, podemos concluir diretamente que  $(G, k)$  é uma instância SIM. Suponhamos, portanto, que as hipóteses da Afirmação 4.21 e da Afirmação 4.22 sejam ambas falsas, isto é, que  $|K| < (d + 1)k$  e  $|I_h| < 2dk$ . Então temos que

$$|V| = |K| + |I| = |K| + |I_h| + |I_s| < (d + 1)k + 2dk + (d + 1)k < 4(d + 1)k.$$

Desse modo, se a desigualdade anterior não for verdadeira, isto é, se  $|V| \geq 4(d + 1)k$ , então necessariamente pelo menos uma das hipóteses da Afirmação 4.21 ou da Afirmação 4.22 será verdadeira, e em qualquer caso podemos concluir certamente que  $(G, k)$  é uma instância SIM. Consequentemente, nosso *núcleo* linear é extremamente simples: se  $|V| \geq 4(d + 1)k$ , informamos que  $(G, k)$  é uma instância SIM, e caso contrário  $|V| < 4(d + 1)k$ , como queríamos.  $\square$

**Demonstração da Afirmação 4.21:** Mostramos que se  $|K| \geq (d + 1)k$ , então a Regra 6<sup>+</sup> com  $c = 2$  pode ser aplicada pelo menos  $k$  vezes, implicando que  $(G, k)$  é uma instância SIM. De fato, caso tenhamos  $|K| \geq d + 1$ , será possível aplicar o seguinte procedimento: seja  $v$  um vértice qualquer de  $K$ , e  $u_1$  um vizinho de  $v$  em  $I$ , que existe por hipótese. Uma vez que  $u_1$  tem grau no máximo  $d$  em  $G$ , e  $|K| \geq d + 1$  necessariamente existe um vértice  $u_2 \in K$  tal que  $\{v, u_2\} \in E$  e  $\{u_1, u_2\} \notin E$ . Note que devido à hipótese de  $G$  ser conexo,  $G - \{v, u_1, u_2\}$  é também conexo. Portanto a Regra 6<sup>+</sup> para  $c = 2$  pode ser aplicada a  $\{v, u_1, u_2\}$ . Então removemos de  $G$  o conjunto de vértices  $N[u_1] \cup \{u_2\}$ , que tem tamanho no máximo  $d + 1$ . Note que o grafo resultante ainda é  $d^*$ -split e que nessa etapa de redução do tamanho de  $G$  tem-se  $k' = k - 1$ . Como  $|K| \geq (d + 1)k$ , este procedimento pode ser aplicado pelo menos  $k$  vezes, e a afirmação é válida.

**Demonstração da Afirmação 4.22:** Mostramos que se  $|I_h| \geq 2dk$ , então é possível aplicar iterativamente a Regra 6<sup>+</sup> à instância  $(G, k)$  até que se obtenha uma instância  $(G', k')$  com  $k' \leq 0$ , o que implica em  $G$  ser uma instância SIM.

Aplicamos a  $(G, k)$  o seguinte procedimento. Faça  $i := 1$ ,  $G^1 := G$ ,  $K_h^1 := K_h$ ,  $I_h^1 := I_h$ , e  $k_1 := k$ . Proceda do seguinte modo:

1. Seja  $v_i$  um vértice qualquer de  $K_h^i$ , faça  $N^i := N_{I_h^i}(v_i)$ , e  $c_i := |N^i|$ . Note que  $c_i \geq 2$  e que  $G^i - N^i$  é conexo, pois estamos removendo uma parte do conjunto independente composta apenas por vértices que são adjacentes a  $v_i$ , isto é, os vértices de  $I_h^i - N^i$  são adjacentes a vértices de  $K_h^i - \{v_i\}$  e  $K_h^i$  é uma clique.
2. Aplique a Regra 6<sup>+</sup> a  $\{v_i\} \cup N^i$ .
3. Seja  $D_K^i$  o conjunto de vértices de  $K_h^i$  que têm no máximo um vizinho em  $I_h^i \setminus N^i$ , faça  $D_I^i := N_{I_h^i \setminus N^i}(D_K^i)$ ; veja a Figura 4.15 como um exemplo desses

conjuntos para  $d = 3$ . Como cada vértice de  $K_h^i$  tem pelo menos dois vizinhos em  $I_h^i$  e cada vértice de  $I_h^i$  tem grau no máximo  $d$ , segue que  $|D_K^i| \leq (d-1)|N^i| = (d-1)c_i$ . Por outro lado, de acordo com a definição, vale que  $|D_I^i| \leq |D_K^i| \leq (d-1)c_i$ .

4. Faça  $G^{i+1} := G^i - (\{v_i\} \cup N^i \cup D_K^i \cup D_I^i)$ ,  $K_h^{i+1} := K_h^i \setminus (\{v_i\} \cup D_K^i)$ ,  $I_h^{i+1} := I_h^i \setminus (N^i \cup D_I^i)$ , e  $k_{i+1} := k_i - c_i + 1$ . Note que  $|I_h^i| - |I_h^{i+1}| = |N^i| - |D_I^i| \leq c_i + (d-1)c_i = dc_i$ , e por construção cada vértice de  $K_h^{i+1}$  tem pelo menos dois vizinhos em  $I_h^{i+1}$ .
5. Se  $|I_h^{i+1}| > 0$ , atualize  $i \leftarrow i + 1$  e retorne ao Passo 1. Caso contrário, pare o procedimento.

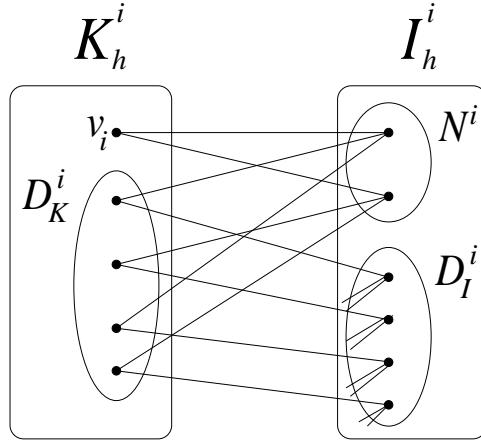


Figura 4.15: Exemplo para  $d = 3$  dos conjuntos definidos na demonstração da Afirmação 4.22.

Assuma que o procedimento acima tenha sido aplicado  $p \geq 1$  vezes, e faça  $k' := k_{p+1}$ . Note que  $k' = k - \sum_{i=1}^p (c_i - 1)$ . Como  $k'$  é o parâmetro obtido de  $k$  através da aplicação iterativa da Regra 6<sup>+</sup> em  $G$  aos conjuntos descritos no Passo 2 acima, nosso objetivo é mostrar que  $k' \leq 0$ .

De acordo com a condição no Passo 5 acima, necessariamente  $|I_h^{p+1}| = 0$ . Uma vez que para todo  $1 \leq i \leq p$  temos que, (como foi discutido no Passo 4 acima),  $|I_h^i| - |I_h^{i+1}| \leq dc_i$ , é possível usar a hipótese  $|I_h^1| = |I_h| \geq 2dk$  e concluir que

$$2dk \leq |I_h^1| = (|I_h^1| - |I_h^2|) + (|I_h^2| - |I_h^3|) + \dots + (|I_h^p| - |I_h^{p+1}|) = \sum_{i=1}^p (|I_h^i| - |I_h^{i+1}|) \leq d \cdot \sum_{i=1}^p c_i,$$

Portanto, segue que  $\sum_{i=1}^p c_i \geq 2k$ . Como, para todo  $1 \leq i \leq p$  vale  $c_i \geq 2$  (veja o Passo 1 acima) e a função  $x \mapsto x - 1$  é maior ou igual à função  $x \mapsto x/2$  para  $x \geq 2$ ,

temos que  $\sum_{i=1}^p (c_i - 1) \geq \sum_{i=1}^p c_i / 2 \geq k$ . Finalmente,

$$k' = k - \sum_{i=1}^p (c_i - 1) \leq k - k = 0,$$

e portanto podemos concluir que  $(G, k)$  é uma instância SIM, como havíamos afirmado.

Os resultados desse capítulo datam de 2015 e encontram-se no artigo “*Improved Kernels for Signed Max Cut Parameterized Above Lower Bound on  $(r, \ell)$ -Graphs*” [30] que teve a colaboração do pesquisador Ignasi Sau. Em dezembro de 2016 foi apresentado no ISAAC 2016 o artigo “*Linear Kernels and Linear Time Algorithms for Finding Large Cuts*” de Michael Etscheid e Matthias Mnich [28] no qual é determinado um núcleo linear para grafos em geral, melhorando consideravelmente tanto os resultados de Crowston, Gutin, Jones e Muciaccia [15], bem como os nossos resultados. Neste artigo, Etscheid e Mnich utilizam a definição de classes de grafos  $\lambda$ -*extendíveis*, criada por Poljak e Turzik [54] para definir classes *fortemente*  $\lambda$ -*extendíveis*. Etscheid e Mnich mostraram que a classe dos grafos bipartidos é fortemente  $\frac{1}{2}$ -*extendível* e que a classe dos grafos fortemente  $\lambda$ -*extendíveis* está contida na classe dos grafos  $\lambda$ -*extendíveis*. A partir daí utilizaram o resultado de Poljak e Turzik de que dado um grafo conexo  $G$  arbitrário e uma classe de grafos  $\mathcal{C}$   $\lambda$ -*extendível*, existe um subgrafo  $H$  de  $G$  tal que  $H \in \mathcal{C}$  e  $|E(H)| \geq \lambda|E(G)| + \frac{1-\lambda}{2}(|V(G)| - 1)$ . Desse modo, para SIMPLE MAX CUT em um grafo  $G$  com  $|V(G)| = n$  e  $|E(G)| = m$  obtém-se sempre um subgrafo bipartido com pelo menos  $\frac{m}{2} + \frac{n-1}{4}$  arestas, isto é,  $mc(G) \geq \frac{m}{2} + \frac{n-1}{4}$ , sendo esse último número o já conhecido valor  $pt(G)$ . Em seguida, Etscheid e Mnich modificaram as regras de redução do tipo *two-way* aplicadas por Crowston, Gutin, Jones e Muciaccia de modo a garantir que a aplicação dessas regras ao grafo  $G - S$  não aumenta o número de componentes conexas no grafo resultante, ou seja, a aplicação das regras não “quebra” as componentes conexas de  $G - S$  no grafo resultante  $G' - S$ . Desse modo, o mesmo conjunto  $S$  de vértices marcados no grafo original  $G$  pode ser visto como conjunto de vértices marcados no grafo resultante  $G'$  após a aplicação exaustiva das regras de redução. Com isso é possível remover juntamente com cada vértice  $v \in S$  um subconjunto formado por vértices adjacentes a  $v$ , cada um no interior de um bloco de  $G' - S$ , determinando uma estrutura  $Y_v$  do tipo *estrela* que ao ser removida provoca um crescimento linear do parâmetro  $k$  em  $\frac{m}{2} + \frac{n-1}{4} + k$  com relação a  $|Y_v|$ . A partir disso, é possível fazer a remoção das estruturas  $Y_v$  de forma que os grafos obtidos a cada passo sejam ainda conexos, o que provoca um crescimento do parâmetro  $k$  da magnitude da soma de  $|Y_v|$  para os  $v \in S$  removidos.

# Capítulo 5

## Cortes de Areias em Grafos Aresta-Coloridos

### 5.1 Introdução

Vamos relembrar alguns conceitos já introduzidos no Capítulo 1 e que serão necessários ao longo desse capítulo. Seja  $G = (V, E)$  um grafo conexo simples, e sejam  $S$  e  $T$  dois subconjuntos não vazios disjuntos de  $V(G)$ . Usamos a notação  $[S, T]$  para representar o subconjunto de  $E(G)$  formado pelas arestas que possuem um extremo em  $S$  e o outro em  $T$ . Um *corte de arestas* de  $G$  é um subconjunto  $[S, V \setminus S] = \partial S$  de  $E(G)$  para algum subconjunto não vazio  $S \subsetneq V(G)$ . Dados  $s$  e  $t$  dois vértices distintos de  $V(G)$ , quando queremos destacar a importância desses vértices no grafo  $G$ , diremos que  $G$  é um  $(s, t)$ -grafo. Um  $(s, t)$ -corte de  $G$  é um corte de arestas  $[S, T]$  de  $G$  tal que  $s \in S$  e  $t \in T$ . Um  $(s, t)$ -caminho é um caminho ligando os vértices  $s$  e  $t$ .

Seja  $G^c = (V, E, c)$  um grafo conexo simples  $G = (V, E)$  com uma coloração de arestas  $c : E(G) \rightarrow \{1, 2, \dots, p\}$  não necessariamente própria, isto é, arestas incidentes a um mesmo vértice podem ter a mesma cor. Aqui os rótulos numéricos representam as cores atribuídas às arestas de  $G$  através da coloração  $c$ . O *número de cores*  $C(\partial S)$  de um corte de arestas  $\partial S$  é o **número de cores** presentes nesse corte. Denotamos por  $c(\partial S) = \{c(e) | e \in \partial S\}$  o conjunto formado pelas cores presentes no corte de arestas  $\partial S$ . Observe que o número de cores do corte é a cardinalidade do conjunto  $c(\partial S)$ , ou seja,  $C(\partial S) = |c(\partial S)|$ .

A partir de agora consideramos  $G^c = (V, E, c)$  um grafo conexo simples munido de uma coloração de arestas  $c$ . Dado um grafo  $G^c$ , nosso primeiro objetivo ao longo desse capítulo é determinar cortes de arestas que tenham a menor capacidade possível e outros que tenham a maior capacidade possível. O estudo dos proble-

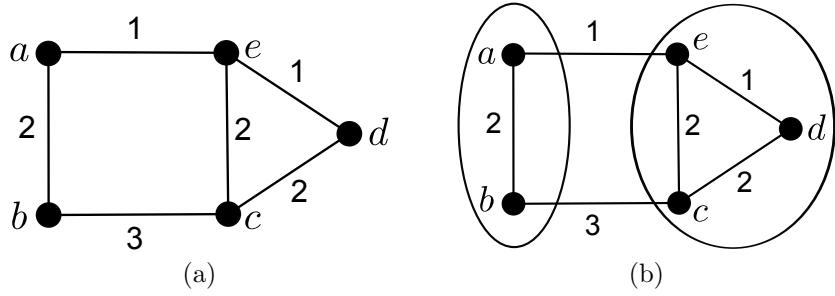


Figura 5.1: (a) Um grafo aresta-colorido  $G$  e (b) Um corte  $\partial S$  de  $G$  com  $S = \{a, b\}$  e capacidade  $C(\partial S) = 2$ .

mas de cortes de arestas em grafos aresta-coloridos nos foi sugerido pelo professor Uéverton dos Santos Souza durante o Exame de Qualificação. O problema de otimização que consiste na determinação do corte de arestas colorido com a menor capacidade possível é chamado de MIN-COLORED-CUT. A seguir apresentamos a forma de decisão desse problema.

MIN-COLORED-CUT

INSTÂNCIA: Um grafo aresta-colorido  $G^c$  e um inteiro  $k > 0$ .

PERGUNTA: Existe um subconjunto não vazio  $S \subsetneq V(G)$  tal que  $C(\partial S)$  é no máximo  $k$ ?

Associado ao problema MIN-COLORED-CUT, temos o problema de decisão MIN-COLORED- $(s, t)$ -CUT, obtido quando fixamos dois vértices  $s \in S$  e  $t \in T$ .

MIN-COLORED- $(s, t)$ -CUT

INSTÂNCIA: Um grafo aresta-colorido  $G^c$  com vértices fixados  $s, t \in V$  e um inteiro  $k > 0$ .

PERGUNTA: Existe um corte de arestas  $[S, T]$  com  $s \in S$ ,  $t \in T$  e capacidade no máximo  $k$ ?

Os problemas MIN-COLORED-CUT e MIN-COLORED- $(s, t)$ -CUT foram inicialmente introduzidos por Rizzi, Coudert, Datta, Perennes, Rivano e Voge [11, 57] que utilizaram grafos aresta-coloridos no estudo de Grupos de Recursos de Risco Compartilhado (*Shared Risk Resource Group (SRRG)*). Em particular, estudaram os problemas MC-CUT e MC- $(s, t)$ -CUT que correspondem aos problemas MIN-COLORED-CUT e MIN-COLORED- $(s, t)$ -CUT, respectivamente. Definiram o conceito de *span* de uma cor (número de componentes conexas do grafo formadas por arestas coloridas com aquela cor) e provaram que esses problemas são polinomiais para grafos aresta-coloridos com span no máximo um. Mostraram também que esses problemas são NP-difíceis ou NP-completos quando o span máximo do grafo é limitado por uma constante  $k > 1$ .

Note que se a função de coloração  $c : E(G) \rightarrow \{1, 2, \dots, p\}$  é injetiva, isto é, se todas as arestas possuem cores distintas, então os problemas MIN-COLORED-CUT e MIN-COLORED- $(s, t)$ -CUT correspondem exatamente a MIN-CUT e MIN- $(s, t)$ -CUT, respectivamente, sendo portanto solucionáveis em tempo polinomial pelo algoritmo clássico do Fluxo Máximo de Ford e Fulkerson [35] que posteriormente foi aprimorado por Edmonds e Karp [22].

Não podemos resolver MIN-COLORED-CUT através do algoritmo anterior porque nesse caso o tamanho do corte de arestas geralmente não importa: podemos ter um corte de arestas com “muitas” arestas coloridas com poucas cores e, por outro lado podemos encontrar um corte de arestas “pequeno” que apresenta muitas das cores da coloração de arestas desse grafo. A Figura 5.2 ilustra dois exemplos dessas situações num mesmo grafo.

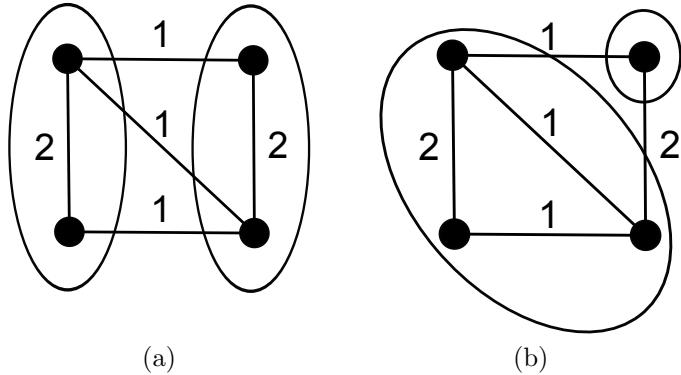


Figura 5.2: (a) Um corte colorido mínimo com as partes circuladas e que apresenta muitas arestas(3) e poucas cores(1) e (b) um corte colorido máximo de um grafo com muitas cores(2) e poucas arestas(2).

Por outro lado, podemos estar interessados em encontrar cortes de arestas com a maior capacidade. Este problema de otimização é chamado MAX-COLORED-CUT. A seguir apresentamos a forma de decisão do problema.

MAX-COLORED-CUT

INSTÂNCIA: Um grafo aresta-colorido  $G^c$  e um inteiro  $k > 0$ .

PERGUNTA: Existe um subconjunto não vazio  $S \subsetneq V(G)$  tal que  $C(\partial S)$  é pelo menos  $k$ ?

Esse problema é uma generalização do problema clássico do corte máximo, SIMPLE MAX-CUT [39] e portanto também é NP-completo. O problema SIMPLE MAXCUT corresponde ao caso em que cada par de arestas de  $G$  está colorido com cores distintas, isto é, cada aresta possui sua cor exclusiva e consequentemente a capacidade de um corte colorido de arestas é exatamente o número de arestas presentes nesse corte.

Cabe também perguntar se existe algum corte de arestas do grafo  $G^c$  com todas

as cores da coloração  $c$ . Este problema foi sugerido pelo professor Carlos Alberto de Jesus Martinhon. Caso exista, esse é um corte colorido máximo. Esse problema é chamado **COLORFUL CUT** e constitui um caso particular de **MAX-COLORED-CUT**. Por exemplo, se considerarmos grafos completos  $K_n$  com  $n \geq 3$  e uma coloração de arestas própria, então a resposta a **COLORFUL CUT** é sempre NÃO. Em grafos bipartidos, a resposta é sempre SIM, independentemente da coloração de arestas do grafo. A seguir apresentamos **COLORFUL CUT** na forma de decisão.

**COLORFUL CUT**

INSTÂNCIA: Um grafo aresta-colorido  $G^c$ .

PERGUNTA: Existe um subconjunto não vazio  $S \subsetneq V(G)$  tal que  $C(\partial S) = |c(G)|$ ?

Neste capítulo estudamos a complexidade computacional desses problemas, com ênfase especial em suas complexidades aproximativa e parametrizada para várias escolhas de parâmetros. Para uma introdução ao assunto veja [17, 21, 33, 51]. Ao longo do capítulo, denotamos por  $n$  o número de vértices do grafo de entrada do problema considerado.

## 5.2 Preliminares

A seguir apresentamos um *framework* para demonstrar a inviabilidade de núcleos polinomiais em um problema FPT. Para estabelecê-lo precisaremos da definição de *OU-composição*.

Um *algoritmo de OU-composição* para um problema parametrizado  $\Pi \subseteq \Sigma^* \times \mathbb{N}$  é um algoritmo que:

1. recebe como entrada uma sequência de instâncias  $(G_1, k_1), \dots, (G_t, k_t)$  de  $\Pi$  e
2. produz, em tempo polinomial em  $\Sigma|G_i|$ , uma instância  $(G, k)$  tal que
3.  $k \leq f(\max |k_i|)$  com  $f$  uma função polinomial e
4.  $(G, k)$  é uma instância SIM se e somente se existe  $1 \leq i \leq t$  tal que  $(G_i, k_i)$  é uma instância SIM.

O Teorema 5.1 dado a seguir nos fornece condições para determinar quando um problema parametrizado **não** admite núcleo polinomial.

**Teorema 5.1.** [5] *Se o problema  $\Pi$  admite um algoritmo de OU-composição, então  $\Pi$  não admite núcleo polinomial a menos que  $\text{NP} \subseteq \text{coNP/poly}$ .*

### 5.3 Resultados de NP-completude para o corte colorido mínimo

Richard Karp apresentou 21 problemas combinatórios em seu artigo de 1972 e mostrou que todos eles são NP-completos. Entre esses problemas figura SET COVER. A seguir apresentamos o problema SET COVER em sua forma de decisão.

SET COVER

INSTÂNCIA: Um conjunto  $C = \{1, \dots, m\}$ , uma família  $S = \{S_1, \dots, S_p\}$  de subconjuntos de  $C$  tal que  $C \subset \bigcup_{i=1}^p S_i$  e um inteiro  $k > 0$ .

PERGUNTA: Existe uma subfamília  $S_{i_1}, \dots, S_{i_k}$  com  $k$  conjuntos de  $S$  tal que  $C \subset \bigcup_{j=1}^k S_{i_j}$ ?

Se  $C \subset \bigcup_{j=1}^k S_{i_j}$  dizemos que a subfamília  $S_{i_1}, \dots, S_{i_k}$  *cobre*  $C$  ou que  $S_{i_1}, \dots, S_{i_k}$  é uma *cobertura do conjunto*  $C$ .

Apresentamos a seguir uma redução de instâncias de SET COVER em instâncias de MIN-COLORED- $(s, t)$ -CUT, mostrando que esse último problema é NP-completo.

**Teorema 5.2.** MIN-COLORED- $(s, t)$ -CUT é *NP-completo*.

*Demonstração.* MIN-COLORED- $(s, t)$ -CUT está em NP porque dada uma instância  $(G^c, k)$  de MIN-COLORED- $(s, t)$ -CUT, podemos considerar um certificado como sendo um subconjunto  $S \subset V$  com  $|S| \leq k$  e podemos checar se existem  $k$  cores no corte  $[S, V \setminus S]$  em tempo  $O(m)$ .

Seja  $((C, S), k)$  uma instância do problema set cover. Nós vamos construir em tempo polinomial no tamanho de  $((C, S), k)$  uma instância  $(G^c = (V, E, c), s, t, k')$  de MIN-COLORED- $(s, t)$ -CUT tal que existe uma cobertura  $S_{i_1}, \dots, S_{i_k}$  de  $C$  com no máximo  $k$  conjuntos de  $S$  se e somente se existe um  $(s, t)$ -corte  $[S, V \setminus S]$  de  $G^c = (V, E, c)$  com no máximo  $k'$  cores.

Vamos identificar os elementos de  $C$  e os subconjuntos de  $S$  com vértices de um grafo bipartido  $G' = (V', E')$  de modo que  $V' = C \cup S$ ,  $C$  e  $S$  são conjuntos independentes e existe uma aresta unindo um elemento  $i$  de  $C$  a um conjunto  $S_j$  de  $S$  se  $i \in S_j$ . A Figura 5.3(a) ilustra um exemplo do grafo  $G'$ .

A partir de  $G'$  construimos o grafo  $G^c = (V, E, c)$  do seguinte modo:

- Crie dois vértices  $s$  e  $t$ .

- Para cada  $i \in C$  com  $N_{G'}(i) = \{S_i^1, S_i^2, \dots, S_i^{d(i)}\}$  crie  $P_i = (s, p_i^1, p_i^2, \dots, p_i^{d(i)-1}, t)$  um caminho adicional unindo  $s$  a  $t$  de forma que o número de arestas em  $P_i$  seja igual ao grau de  $i$  em  $G'$ , e cada aresta em  $P_i$  tenha uma cor associada a um conjunto  $S_j$  vizinho de  $i$  em  $G'$ , isto é,  $c(\{s, p_i^1\}) = S_i^1, c(\{p_i^j, p_i^{j+1}\}) = S_i^j$  para  $1 \leq j \leq d(i)-2$  e  $c(\{p_i^{d(i)-1}, t\}) = S_i^{d(i)}$ .

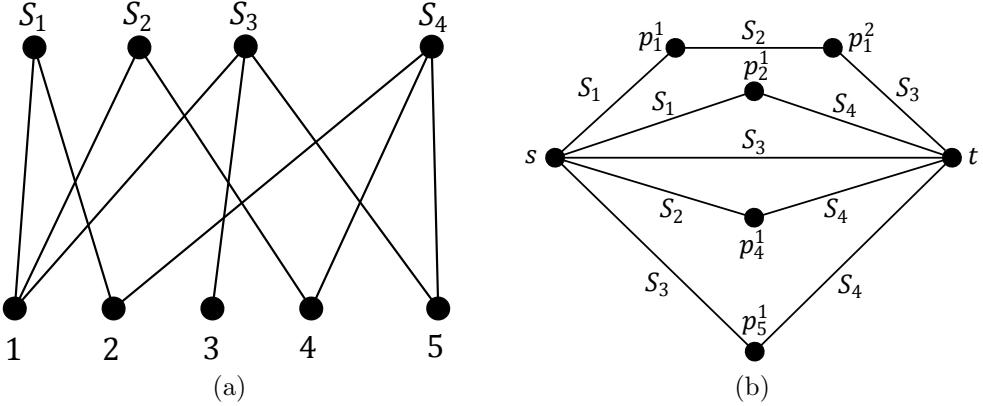


Figura 5.3: (a) Grafo bipartido  $G'$  obtido a partir da instância  $((C, S), k) = ((\{1, 2, 3, 4, 5\}, \{S_1, S_2, S_3, S_4\}), 2) = ((\{1, 2, 3, 4, 5\}, \{\{1, 2\}, \{1, 4\}, \{1, 3, 5\}, \{2, 4, 5\}\}), 2)$  usado na redução de SET COVER e (b) o  $(s, t)$ -grafo  $G^c$ . Veja que existe uma cobertura de conjuntos  $S_3, S_4$  de tamanho 2 para  $C = \{1, 2, 3, 4, 5\}$  se, e somente se, existe um  $(s, t)$ -corte de  $(G^c, s, t, k' = 2)$  contendo as  $k' = k = 2$  cores  $S_3$  e  $S_4$ .

Para a conveniência do leitor, a Figura 5.3 ilustra a construção de  $G^c$  a partir do grafo  $G'$ .

Suponha que o  $(s, t)$ -grafo  $G^c$  possui um  $(s, t)$  corte de arestas  $[S, T]$  com capacidade  $k$  ( $k$  cores) em que  $s \in S$  e  $t \in T$ . Seja  $\{S_1, S_2, S_3, \dots, S_k\}$  o conjunto das cores do corte  $[S, T]$ . Como  $G$  tem  $m$  caminhos disjuntos ligando  $s$  a  $t$ , qualquer  $(s, t)$ -corte contém pelo menos uma aresta de cada  $(s, t)$ -caminho. Logo, por definição de  $[S, T]$ , cada um dos  $m$   $(s, t)$ -caminhos possui pelo menos uma aresta colorida com uma das  $k$  cores do corte. Assim, para cada  $i \in \{1, 2, 3, \dots, m\}$  existe  $S_j, j \in \{1, 2, 3, \dots, k\}$  tal que  $i \in S_j$ . Portanto,  $C \subset \bigcup_{j=1}^k S_j$ .

Por outro lado, suponha que existem  $k$  conjuntos  $S_{i_1}, S_{i_2}, \dots, S_{i_k}$  de  $S$  cobrindo  $C$ , i.e.,  $C \subset \bigcup_{j=1}^k S_{i_j}$ . Por construção, existe pelo menos um representante de cada uma das cores  $S_{i_1}, S_{i_2}, \dots, S_{i_k}$  em cada um dos  $(s, t)$ -caminhos de  $G'$ . Dessa forma, o  $(s, t)$ -grafo  $G^c$  tem um corte de arestas  $[S, T]$  com  $s \in S$  e  $t \in T$  com capacidade no máximo  $k$ .  $\square$

Dado um grafo  $G = (V, E)$ , um subconjunto  $C \subset V$  é dito uma *cobertura de vértices* (*vertex cover*) de  $G$  se toda aresta de  $G$  possui pelo menos um extremo em  $C$ . A seguir enunciamos o problema de decisão VERTEX COVER.

VERTEX COVER

INSTÂNCIA: Um grafo  $G = (V, E)$  e um inteiro  $k > 0$ .

PERGUNTA: Existe uma cobertura de vértices  $S \subseteq V$  com tamanho no máximo  $k$ ?

O resultado a seguir é devido a Garey, Johnson e Stockmeyer [39].

**Lema 5.3.** [39]. VERTEX COVER permanece NP-completo mesmo restrito a grafos cúbicos.

Observe que o problema VERTEX COVER é um caso particular de SET COVER quando os conjuntos  $S_j$  da família  $S$  representam as vizinhanças fechadas de cada vértice em  $V(G)$  e os elementos  $i \in C$  representam as arestas em  $E(G)$ . Portanto a construção da instância para o Teorema 5.4 será análoga à realizada na Seção 5.3.

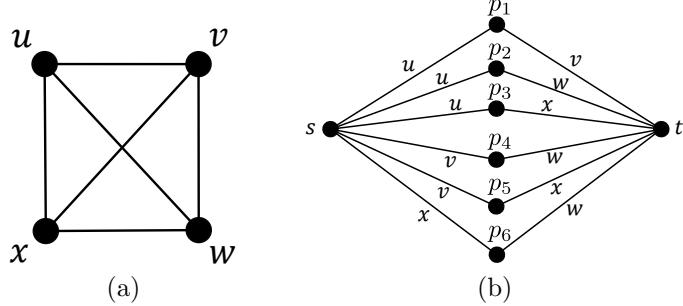


Figura 5.4: (a) O grafo cúbico  $G$  e (b) seu  $(s, t)$ -grafo  $G^c$ .

**Teorema 5.4.** MIN-COLORED- $(s, t)$ -CUT é  $NP$ -completo mesmo se toda cor aparece no máximo três vezes e cada  $(s, t)$ -caminho tem tamanho dois.

*Demonstração.* MIN-COLORED- $(s, t)$ -CUT está em  $NP$  porque dada uma instância  $(G^c, k)$  de MIN-COLORED- $(s, t)$ -CUT, podemos considerar um certificado como sendo um subconjunto  $S \subset V$  de  $V$  com  $|S| \leq k$  e podemos checar se  $s \in S$ ,  $t \in V \setminus S$ , e se existem  $k$  cores no corte  $[S, V \setminus S]$  tudo em tempo  $O(m)$ .

Consideremos uma instância de VERTEX COVER dada por  $(G, k)$ , em que  $G = (V, E)$  é um grafo cúbico. Vamos construir em tempo polinomial no tamanho de  $(G, k)$  uma instância  $(G^c, s, t, k')$  de MIN-COLORED- $(s, t)$ -CUT tal que existe uma cobertura de vértices  $S \subset V$  com  $|S| \leq k$  se e somente se existe um  $(s, t)$  corte  $[S, V \setminus S]$  de  $G^c$  com no máximo  $k'$  cores.

Para isso consideramos dois vértices adicionais  $s$  e  $t$  em  $V(G')$  e para cada aresta  $\{u, v\} \in G$  associamos um caminho  $P_i = (s, p_i, t)$  em  $G'$  com  $c(\{s, p_i\}) = u$  e  $c(\{p_i, t\}) = v$ . Finalmente fixamos  $k' = k$ . Para a conveniência do leitor, oferecemos na Figura 5.4 um exemplo de construção.

Tomando um  $(s, t)$ -corte de  $G^c$  com no máximo  $k = k'$  cores, podemos observar que cada um dos  $m$   $(s, t)$ -caminhos tem uma aresta com uma das  $k' = k$  cores no corte, caso contrário  $t$  poderia ser alcançado a partir de  $s$  através de um caminho que não tivesse arestas no corte. Como as arestas de  $G^c$  estão coloridas pelos vértices de  $G$ , e os caminhos de  $G^c$  representam as arestas de  $G$ , então para cada aresta de  $G$  ao menos um de seus extremos possui sua cor no corte, sendo assim as cores presentes no corte induzem uma cobertura de vértices com no máximo  $k$  vértices de  $G$ .

Reciprocamente, se existe uma cobertura de vértices  $B \subset V$  de  $G$  com  $|B| \leq k$ , então todas as arestas de  $G$  têm pelo menos um extremo em  $B$ . Portanto, todos

os  $(s, t)$ -caminhos de  $G^c$  têm pelo menos uma aresta colorida por um vértice de  $B$ , assim podemos escolher um subconjunto dessas arestas para formar um  $(s, t)$ -corte de  $G^c$ .

Observe que o fato de  $G$  ser cúbico, implica que em  $G^c$  cada cor ocorrerá exatamente três vezes, uma vez em cada caminho no qual está presente.  $\square$

A seguir provamos que MIN-COLORED- $(s, t)$ -CUT é NP-completo mesmo se toda cor aparece no máximo duas vezes e cada  $(s, t)$ -caminho tem comprimento no máximo três. Usaremos uma redução de VERTEX COVER em que todo vértice possui grau três e toda aresta incide a no máximo um vértice de grau três.

**Lema 5.5.** VERTEX COVER permanece NP-completo mesmo em grafos com grau máximo três, em que toda aresta incide a no máximo um vértice de grau três.

*Demonstração.* Seja  $G$  um grafo cúbico. Construiremos a partir de  $G$  um grafo  $G'$  de grau máximo três, em que toda aresta incide a no máximo um vértice de grau três, de modo que  $G$  possua uma cobertura de tamanho  $k$  se e somente se  $G'$  possuir uma cobertura de tamanho  $k' = k + |E(G)|$ . A construção é dada a seguir partindo de um grafo  $G'$  vazio. Tal construção é ilustrada na Figura 5.5.

- Faça  $V(G') = V(G) \cup \{v_e^1, v_e^2 : e \in E(G)\}$ ;
- Para cada aresta  $e = (u, w) \in E(G)$  acrescente em  $G'$  o caminho  $(u, v_e^1, v_e^2, w)$ .

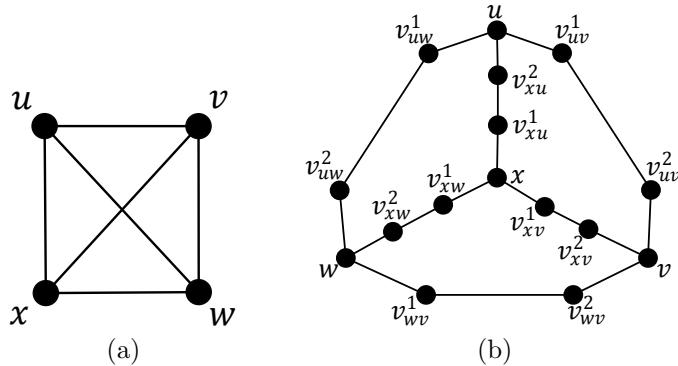


Figura 5.5: (a) Um grafo cúbico  $G$  e (b) seu grafo associado  $G'$  para o Lema 5.5.

Seja  $B$  uma cobertura por vértices de  $G$ , tal que  $|B| = k$ . Podemos construir uma cobertura  $B'$  de  $G'$  atribuindo todo vértice em  $B$  também a  $B'$ . Como  $B$  é uma cobertura de  $G$ , então para todo caminho  $(u, v_e^1, v_e^2, w)$  de  $G'$  ( $u, w \in V(G)$ ), temos que  $u$  ou  $w$  ou ambos pertencem a  $B'$ , de acordo com a atribuição que foi feita. Desta forma, se  $u \in B'$  adicionamos  $v_e^2$  em  $B'$ , senão adicionamos  $v_e^1$  em  $B'$ . Sendo assim temos que  $B'$  cobre todas as arestas de  $G'$  e possui tamanho  $k + |E(G)|$ .

Por outro lado, seja  $B'$  uma cobertura de vértices de  $G'$  de tamanho no máximo  $k + |E(G)|$ . Todo caminho  $(u, v_e^1, v_e^2, w)$  de  $G'$  possui ao menos dois vértices em

$B'$ , sendo pelo menos um deles em  $\{v_e^1, v_e^2\}$ . Se  $v_e^1$  e  $v_e^2 \in B'$  podemos atualizar outra cobertura de vértices  $B'$  com  $|B'| \leq |B|$  substituindo  $v_e^1$  e  $v_e^2$  por  $v_e^1$  e  $v$  em  $B$ . Assim, podemos assumir que  $B'$  possui exatamente um vértice em  $\{v_e^1, v_e^2\}$  para todo  $e \in E(G)$ . Sendo assim, podemos definir uma cobertura  $B$  para  $G$  como  $B = V(G) \cap B'$ . Logo  $B$  é uma cobertura de  $G$  de cardinalidade menor ou igual a  $k$ , pois  $B'$  possui exatamente um vértice em  $\{v_e^1, v_e^2\}$  e consequentemente ao menos um vértice em  $\{u, w\}$  para todo  $e = (u, w) \in E(G)$ .  $\square$

**Teorema 5.6.** MIN-COLORED- $(s, t)$ -CUT é NP-completo mesmo se toda cor aparece no máximo duas vezes e cada  $(s, t)$ -caminho tem comprimento no máximo três.

*Demonstração.* Dado um grafo  $G = (V, E)$  em que o grau dos vértices é no máximo três e toda aresta incide a no máximo um vértice de grau três, podemos usar o resultado do Lema 5.5 para mostrar a NP-completude de MIN-COLORED- $(s, t)$ -CUT em  $(s, t)$ -grafos nos quais cada cor aparece no máximo em duas arestas e cada  $(s, t)$ -caminho tem comprimento no máximo três. Para isso, construímos um grafo  $G'$  como no Teorema 5.4 e na Figura 5.4, de modo que a cada aresta  $e = \{u, v\}$  de  $G$  corresponde um  $(s, t)$ -caminho  $P_e = (s, p_e, t)$  em  $G'$  com  $c(\{s, p_e\}) = u$  e  $c(\{p_e, t\}) = v$ . Portanto, se um vértice  $u \in G$  tem grau no máximo três, podemos concluir que em  $G'$  existem no máximo três  $(s, t)$ -caminhos de comprimento dois, cada um deles apresentando a cor  $u$  em uma de suas arestas. Podemos supor que  $G$  possui pelo menos um vértice de grau três, caso contrário  $G$  se reduziria a um caminho ou um ciclo, tornando o problema polinomial. Usaremos o grafo  $G'$  para construir um novo grafo  $G''$  do seguinte modo:

1. Escolha uma cor  $c_i$  que ocorre em três  $(s, t)$ -caminhos de  $G'$  e selecione esses caminhos. Pode-se construir  $G'$  de modo que os três  $(s, t)$ -caminhos  $P_{e_j} = (s, p_{e_j}, t)$  selecionados ( $j = 1, 2, 3$ ) nos quais a cor  $c_i$  está presente tenham a propriedade  $c(\{s, p_{e_j}\}) = c_i$ . Em cada um desses caminhos selecionados, substitua o vértice  $p_{e_j}$  por dois novos vértices  $p'_{e_j}$  e  $p''_{e_j}$ , criando a aresta  $\{p'_{e_j}, p''_{e_j}\}$  e atribuindo a essa aresta uma nova cor  $c_{e_j}$ , diferente das cores presentes em  $G'$  e fazendo  $c(\{s, p'_{e_j}\}) = c_i$ ,  $c(\{p''_{e_j}, t\}) = c(\{p_{e_j}, t\})$ . Em seguida, troque as arestas coloridas com a cor  $c_i$  por uma única aresta  $\{s, p_i\}$ , obtendo novos  $(s, t)$ -caminhos de comprimento três da forma  $P'_{e_j} = (s, p_i, p''_{e_j}, t)$  com a coloração de arestas  $c(\{s, p_i\}) = c_i$ ,  $c(\{p_i, p''_{e_j}\}) = c(\{p'_{e_j}, p''_{e_j}\})$  e  $c(\{p''_{e_j}, t\}) = c(\{p_{e_j}, t\})$  (isto faz com que a cor  $c_i$  apareça no máximo duas vezes em  $G''$ );
2. Repita o procedimento anterior para as demais cores que também ocorrem três vezes no grafo corrente (caso seja necessário, troque a ordem das cores no caminho para que as arestas coloridas com a mesma cor sejam adjacentes em  $s$  para poder aplicar o processo);

Como toda aresta é incidente a no máximo um vértice de grau três, o procedimento acima descrito é aplicado no máximo uma vez para cada  $(s, t)$ -caminho de  $G'$ . Sendo assim,  $G''$  é um  $(s, t)$ -grafo no qual cada  $(s, t)$ -caminho tem comprimento no máximo três e cada cor aparece no máximo duas vezes. Para a conveniência do leitor, oferecemos na Figura 5.6 um exemplo ilustrando cada etapa da construção.

Afirmamos que  $G''$  possui um  $(s, t)$ -corte colorido com capacidade no máximo  $k$  se e somente se  $G$  possui uma cobertura de vértices de cardinalidade no máximo  $k$ .

Suponhamos que  $G''$  possui um  $(s, t)$ -corte colorido com capacidade no máximo  $k$ . Sabemos que cada um dos  $(s, t)$ -caminhos de  $G''$  tem uma aresta no corte. Se o corte só envolve arestas coloridas com as cores antigas, isso significa que para cada  $(s, t)$ -caminho de  $G''$  (aresta de  $G$ ) está sendo tomado um extremo de cada aresta de  $G$ : temos uma cobertura de vértices de  $G$  com no máximo  $k$  vértices. Caso o corte envolva arestas coloridas com as cores novas, é possível trocar cada cor nova por uma antiga, já que a remoção de qualquer aresta do  $(s, t)$ -caminho correspondente desconecta  $s$  de  $t$  ao longo desse caminho e assim recaímos no caso anterior, obtendo novamente uma cobertura de vértices de  $G$  com no máximo  $k$  vértices. A recíproca é idêntica à do Teorema 5.4.  $\square$

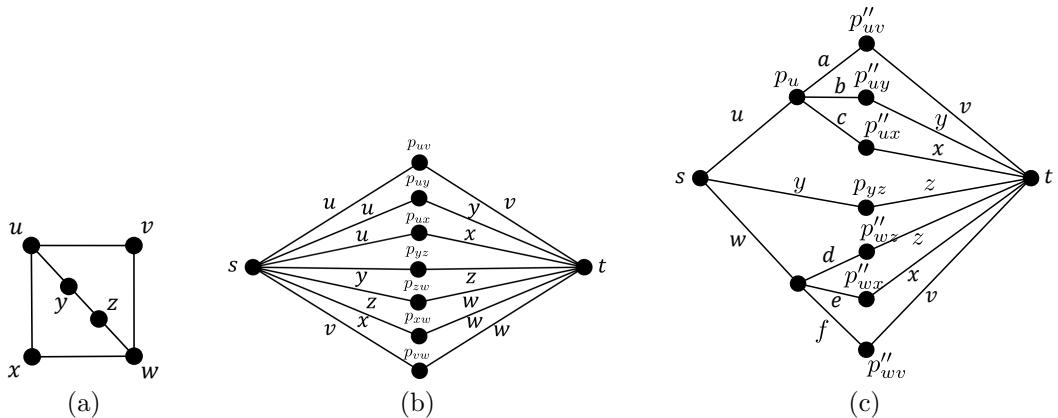


Figura 5.6: (a) O grafo com grau máximo 3,  $G = (V, E)$  no qual toda aresta incide a no máximo um vértice de grau 3, (b) seu  $(s, t)$ -grafo intermediário  $G'$  e (c) o grafo  $G''$  no qual cada cor aparece no máximo 2 vezes e cada  $(s, t)$ -caminho tem comprimento no máximo 3.

Na demonstração acima, embora as cores ocorram no máximo duas vezes, algumas cores aparecem em três  $(s, t)$ -caminhos distintos.

## 5.4 Resultados de complexidade parametrizada para o corte colorido mínimo

Iniciamos esta seção apresentando um resultado de complexidade parametrizada para SET COVER devido a Downey e Fellows [20] que será utilizado para a análise

da complexidade parametrizada de MIN-COLORED- $(s, t)$ -CUT tanto restrito a grafos bipartidos planares quanto restrito a grafos completos.

**Lema 5.7.** [20] SET COVER é  $W[2]$ -completo quando parametrizado pelo tamanho da solução buscada.

**Corolário 5.8.** MIN-COLORED- $(s, t)$ -CUT restrito a grafos bipartidos planares é  $W[2]$ -difícil quando parametrizado pelo tamanho da solução (número de cores presentes na solução).

*Demonstração.* A mesma do Corolário 5.16, utilizando-se o Lema 5.7 e a Proposição 1.2. De fato, observe que o grafo  $H^c$  construído na demonstração do Corolário 5.16 é bipartido planar e, como já mencionamos,  $H^c$  tem um  $(s, t)$ -corte  $[S, T]$  com capacidade  $k$  se e somente se existe um subconjunto de  $S$  de tamanho  $k$  que cobre  $C$ . Tomando  $k$  como parâmetro teremos uma FPT-redução, já que ambos os problemas MIN-COLORED- $(s, t)$ -CUT e SET COVER podem ser parametrizados, respectivamente, pelo tamanho do corte  $k'$  e pelo tamanho da cobertura  $k$ . Além disso, segundo os resultados obtidos ao longo da demonstração do Corolário 5.16, temos o mesmo algoritmo polinomial levando instâncias SIM de  $(C, S, k)$  de SET COVER em instâncias SIM de  $(H^c, k')$  de MIN-COLORED- $(s, t)$ -CUT, com  $k' = k$ . Logo o algoritmo que leva uma instância de SET COVER em MIN-COLORED- $(s, t)$ -CUT restrito a grafos bipartidos planares é uma FPT-redução. Aplicando a Proposição 1.2 concluímos que esses dois problemas pertencem a uma mesma classe da hierarquia  $W$ . Utilizando então o Lema 5.7 concluímos que MIN-COLORED- $(s, t)$ -CUT restrito a grafos bipartidos planares é  $W[2]$ -difícil quando parametrizado pela capacidade do corte colorido.  $\square$

Podemos modificar a redução apresentada na Seção 5.3 também para provar que MIN-COLORED- $(s, t)$ -CUT é  $W[2]$ -difícil em grafos completos.

**Corolário 5.9.** MIN-COLORED- $(s, t)$ -CUT é  $W[2]$ -difícil em grafos completos, quando parametrizado pelo tamanho da solução buscada.

*Demonstração.* Vamos mostrar que MIN-COLORED- $(s, t)$ -CUT parametrizado pela capacidade do corte é  $W[2]$ -difícil em grafos completos a partir de uma redução de SET COVER parametrizado pelo tamanho da cobertura de conjuntos. Construa a partir de uma instância  $(C, S, k)$  de SET COVER (com  $C = \{1, 2, \dots, n\}$ ,  $S = \{S_1, \dots, S_m\}$  e  $k$  o tamanho da cobertura de conjuntos) o  $(s, t)$ -grafo  $(G^c, k)$  como foi feito na Seção 5.3 e, como já vimos, o parâmetro  $k$  é dado pela capacidade do  $(s, t)$ -corte. Caso  $\{s, t\} \in G^c$ , modifique o grafo  $G^c$  substituindo essa aresta pelo caminho  $(s, u, t)$  no qual  $c(\{s, u\}) = c(\{u, t\}) = c(\{s, t\})$ . Crie um grafo completo  $G''$  adicionando arestas que ligam todos os vértices dois a dois não adjacentes de  $G^c$ .

e defina uma coloração de arestas  $c'$  em  $G''$  de modo que  $c'(e) = c(e)$  se  $e \in G$  e  $c'(e) = p + 1$  se  $e \in G'' \setminus G$ , sendo  $p$  o número de cores presentes na coloração de arestas de  $G$ , isto é,  $c : E(G) \rightarrow \{1, \dots, p\}$ . A Figura 5.7 ilustra esses procedimentos. A partir dessa construção teremos o seguinte resultado:  $G''$  tem um  $(s, t)$ -corte de arestas de capacidade no máximo  $k + 1$  se e somente se  $G^c$  tem um  $(s, t)$ -corte de capacidade no máximo  $k$ . Note que se  $G^c$  tem um  $(s, t)$ -corte de capacidade no máximo  $k$ , então o  $(s, t)$ -corte de  $G''$  correspondente apresenta a aresta  $\{s, t\}$  colorida com a cor  $p + 1$ , ou seja,  $G''$  tem um corte com capacidade no máximo  $k + 1$ . Em contrapartida, a aresta  $\{s, t\}$  existe em  $G''$  e tem cor  $c(\{s, t\}) = p + 1$ , já que essa aresta não existia em  $G^c$  (modificado). Logo esta cor está sempre presente em qualquer  $(s, t)$ -corte de arestas de  $G''$ . Desse modo, a cada  $(s, t)$ -corte de arestas de  $G^c$  com capacidade no máximo  $k$ , está associado um  $(s, t)$ -corte de arestas de  $G''$  de capacidade no máximo  $k' = k + 1$ . Como a redução apresentada é uma FPT-redução e SET COVER parametrizado pelo tamanho da cobertura de conjuntos é W[2]-difícil, podemos concluir, usando a Proposição 1.2 que MIN-COLORED- $(s, t)$ -CUT parametrizado pela capacidade do corte é W[2]-difícil em grafos completos.  $\square$

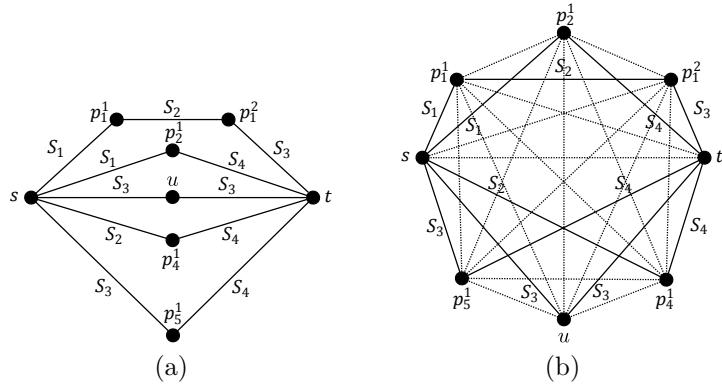


Figura 5.7: (a) O grafo  $G^c$  modificado e (b) o grafo completo  $G''$  no qual as arestas pontilhadas representam as arestas coloridas com a cor  $p + 1$ .

**Teorema 5.10.** MIN-COLORED-CUT e MIN-COLORED- $(s, t)$ -CUT são FPT quando parametrizados pelo número de cores da coloração dada.

*Demonastração.* Dadas as  $p$  cores  $1, \dots, p$  da coloração de arestas do grafo  $G$ , considere  $E_i$  o conjunto formado pelas arestas de  $G$  coloridas com a cor  $i$ , para  $i \in \{1, \dots, p\}$ . Testamos exaustivamente a remoção dos conjuntos  $E_i$  até esgotarmos todas as possibilidades. No pior caso, este procedimento empregaria todas as  $p$  cores. Esse processo tem um custo de  $2^p \cdot O(m)$ , o que justifica o teorema.  $\square$

Observe que MIN-COLORED- $(s, t)$ -CUT pode ser resolvido em tempo polinomial quando  $p = O(\log n)$ . De fato, como vimos na demonstração do Teorema 5.10, o

algoritmo que resolve o problema dado emprega tempo  $2^p \cdot O(m)$ , e quando  $p = O(\log n)$  isso resulta em tempo  $2^{O(\log n)} \cdot O(m) = O(n) \cdot O(m)$  que é polinomial.

Um algoritmo é executado em tempo *pseudo-polynomial* se o seu tempo de execução é polinomial no *valor numérico* da entrada, mas é exponencial no *tamanho* da entrada. Nesta definição, tamanho da entrada significa o espaço alocado para representar a entrada do problema.

Desse modo, MIN-COLORED- $(s, t)$ -CUT pode ser resolvido em tempo pseudo-polynomial quando  $G$  tem um  $(s, t)$ -corte não colorido de tamanho  $O(\log n)$ , pois nesse caso há no máximo  $O(\log n)$  cores presentes nesse corte e portanto será necessário testar todas as combinações com até  $O(\log n)$  cores, sendo o custo total de, no máximo,  $p^{O(\log n)} \cdot O(m)$ .

Como temos um algoritmo FPT para ambos os problemas quando parametrizados pelo número de cores, é interessante verificar se esses problemas possuem núcleo (*kernel*) polinomial. Sendo assim o próximo resultado complementa o Teorema 5.10 acima.

**Lema 5.11.** MIN-COLORED- $(s, t)$ -CUT não admite núcleo polinomial quando parametrizado pelo número de cores, a menos que  $\text{NP} \subseteq \text{coNP/poly}$ .

*Demonstração.* Usando o Teorema 5.1, basta mostrar que o problema  $\Pi = \text{MIN-COLORED-}(s, t)\text{-CUT}$  admite um algoritmo de OU-composição. Para isso, consideremos uma sequência de instâncias  $(G_1, k_1), \dots, (G_q, k_q)$  desse problema, nas quais  $G_1, \dots, G_q$  são grafos conexos dois a dois disjuntos tais que em cada  $G_i$  existem dois vértices distintos  $s_i$  e  $t_i$  fixados e  $k_i$  é um inteiro positivo para  $1 \leq i \leq q$ . Vamos construir  $G$  colapsando os vértices  $t_i$  com  $s_{i+1}$  para cada  $1 \leq i \leq q-1$  e fazendo  $s = s_1$  e  $t = t_q$ . Os grafos  $G_i$  e  $G$  estão ilustrados na Figura 5.8. Em cada grafo  $G_i$  existe uma coloração com as cores  $1, \dots, c_i$  para  $1 \leq i \leq q$ . Tome  $c = \max_{1 \leq i \leq q} c_i$ . Então  $G$  tem uma coloração com  $c$  cores. Observe que  $G$  é construído em tempo polinomial em  $\Sigma|G_i|$  (colapsando os vértices especiais) e  $k \leq \max_{1 \leq i \leq q} c_i$ . Podemos supor, sem perda de generalidade, que  $k_i = k$  para todo  $1 \leq i \leq q$ , já que o problema considerado consiste em determinar, para cada  $G_i$ , se existe um  $(s, t)$ -corte em  $G_i$  com no máximo  $k_i$  cores, o que nos permite tomar  $k = \max k_i$  em todas as instâncias. Desse modo, existe um  $(s, t)$ -corte em  $G$  com no máximo  $k$  cores se e somente se existe algum  $1 \leq i \leq t$  tal que  $G_i$  admite um  $(s_i, t_i)$ -corte com no máximo  $k$  cores. De fato, se algum  $G_i$  admite um corte com no máximo  $k$  cores, este também é um corte de  $G$  com no máximo  $k$  cores. Reciprocamente, se nenhum grafo  $G_i$  admitisse um corte com no máximo  $k$  cores, então todos os  $G_i$  admitiriam cortes com pelo menos  $k+1$  cores e consequentemente  $G$  admitiria apenas cortes com pelo menos  $k+1$  cores. Desse modo o problema admite um algoritmo de OU-composição e o resultado segue.  $\square$

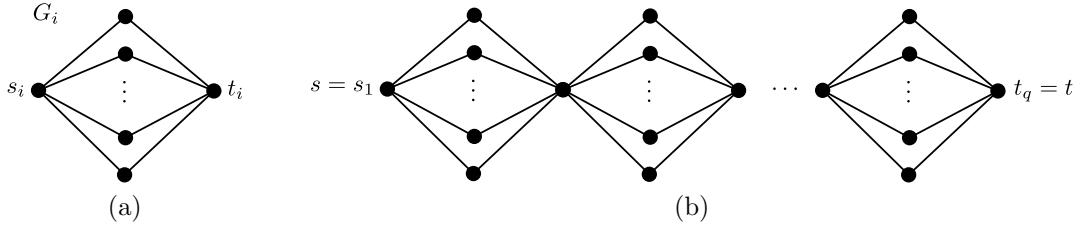


Figura 5.8: (a) Os grafos  $G_i$  e (b) o grafo  $G$  construído no Lema 5.11.

A seguir, algumas definições necessárias são apresentadas. Dado um grafo  $G^c$  aresta-colorido, o *span* de uma cor  $i$  é o número de componentes conexas induzidas pelo conjunto de arestas do grafo coloridas com a cor  $i$ .

Dado um grafo aresta-colorido  $G^c = (V, E, c)$ , se cada cor tem  $span = 1$ , vamos construir o hipergrafo associado no qual o conjunto de vértices que compõem a componente conexa da cor  $i$  formam uma hiperaresta. De acordo com os resultados de Rizzi [57], MIN-COLORED-CUT pode ser resolvido em tempo polinomial nesse caso, já que esse problema é equivalente a determinar o corte de hiperarestas mínimo em hipergrafos.

**Teorema 5.12.** *Dado um grafo aresta-colorido  $G^c$ , denotemos por  $c_2$  o número de cores com span pelo menos dois. O problema MIN-COLORED-CUT pode ser resolvido em tempo  $2^{c_2} \cdot n^{O(1)}$ .*

*Demonstração.* Aplicaremos indução no número de cores com span pelo menos 2. O caso  $c_2 = 0$  pode ser resolvido em tempo polinomial utilizando-se o algoritmo para MIN-CUT em hipergrafos devido a Rizzi [57]. Suponha que existe um algoritmo recursivo que resolve o problema em tempo  $2^{c_2} \cdot n^{O(1)}$  quando  $0 \leq c_2 \leq t$ . Se  $c_2 = t + 1$ , para cada cor  $i$  com span pelo menos 2, consideremos os dois grafos seguintes:  $G_1^i$  (no qual todas as arestas com a cor  $i$  são removidas), e  $G_2^i$  (no qual cada componente com a cor  $i$  recebe uma nova cor diferente das já existentes). Observe que  $c_2(G_1^i) = c_2(G_2^i) = c_2(G^c) - 1$  e isso nos permite aplicar o algoritmo recursivo tanto para  $G_1^i$  como para  $G_2^i$ . Note que  $OPT(G^c) = \min\{OPT(G_1^i) + 1, OPT(G_2^i)\}$ . De fato, temos duas possibilidades a considerar: a cor  $i$  está presente no corte mínimo ou não. Caso a cor  $i$  esteja presente no corte colorido mínimo, devemos remover essa cor e além disso procurar a solução ótima no grafo remanescente  $G_1^i$ . Nesse caso teremos  $OPT(G^c) = OPT(G_1^i) + 1$ . Caso a cor  $i$  não esteja presente no corte colorido mínimo, se considerarmos as componentes conexas dessa cor, cada uma delas colorida com uma nova cor diferente das já existentes em  $G^c$ , podemos observar que nenhuma dessas novas cores está presente no corte ótimo, pois se assim fosse, a remoção da cor  $i$  original já teria tornado o grafo  $G^c$  desconexo. Nesse caso, o surgimento das novas cores não aumenta o tamanho do corte com relação ao grafo original e assim temos  $OPT(G^c) = OPT(G_2^i)$ .

Mais detalhadamente, o algoritmo funciona da seguinte forma: seja  $OPT(G^c)$  o número de cores presentes no corte mínimo colorido para  $G^c$ . Observe que  $OPT(G_1^i) \leq OPT(G^c) \leq OPT(G_2^i)$  para cada  $1 \leq i \leq c_2$ . Seja  $i$  uma cor com span maior que 1. Se existe um corte ótimo de  $G^c$  que não contém  $i$ , então  $OPT(G^c) = OPT(G_2^i)$ . Caso contrário  $OPT(G^c) = OPT(G_1^i) + 1$ .  $\square$

O próximo Lema é devido a Rizzi [57] e considera algoritmos de tempo randomizado polinomial para o problema MIN-COLORED-CUT.

**Lema 5.13.** [57] MIN-COLORED-CUT é solucionável em tempo randomizado polinomial, quando o span máximo da instância  $G^c$  é delimitado por uma constante  $q$ .

Dado um inteiro  $q > 0$ , denotemos por  $c_q(G^c)$  o número de cores com span pelo menos  $q$ .

**Teorema 5.14.** O problema MIN-COLORED-CUT pode ser resolvido por um algoritmo randomizado em tempo  $2^{c_q(G^c)} \cdot n^{O(q)}$ .

*Demonstração.* Basta aplicar o algoritmo apresentado no Teorema 5.12 decrementando a cada passo o span das cores que possuem span maior que  $q$ , até obter na árvore de recursão, grafos em que as cores têm span no máximo  $q$  (caso base da nossa recursão) que será solucionado diretamente pelo algoritmo randomizado polinomial de Blin, Bonizzoni, Dondi, Rizzi e Sikora [3], o qual nada mais é do que o algoritmo clássico de contração de cores. Nesse algoritmo, uma cor é escolhida aleatoriamente no conjunto de cores da coloração de arestas e todas as arestas coloridas com aquela cor são contraídas, dando origem a um novo grafo no qual aquela cor não está mais presente. Nesse processo, cada aresta  $\{u, v\} \in E(G)$  colorida com uma cor  $i$  dá origem a um supervértice  $w \in V(G')$  no novo grafo  $G'$  obtido após a contração. À medida que esse processo recursivo evolui, a cada vértice  $w \in V(G')$  pode ser associado um conjunto  $S(w) \subset V(G)$  formado pelos vértices que desapareceram na contração que deu origem a  $w$ . O algoritmo termina quando é obtido um grafo  $G'$  com apenas dois supervértices  $v_A$  e  $v_B$  que dão origem ao corte colorido  $(A, B)$  no qual  $A = S(v_A)$  e  $B = S(v_B)$ .  $\square$

## 5.5 Resultados de inaproximabilidade para o corte colorido mínimo

Com base nas reduções anteriores e num resultado de Dinur e Steurer [18], citado no Corolário 5.16, vamos concluir que MIN-COLORED- $(s, t)$ -CUT, mesmo restrito a grafos bipartidos planares ou completos não pode ser aproximado por um fator de

$(1 - \epsilon) \ln(p)$  ou  $\frac{(1-\epsilon)}{2} \ln(p)$  para qualquer constante  $\epsilon > 0$ , respectivamente, a menos que  $P = NP$ . Aqui  $p$  representa o número de cores da coloração  $c : E(G) \rightarrow \{1, 2, \dots, p\}$ .

**Lema 5.15.** [18]. *Dada uma instância  $(C, S)$  de SET COVER, na qual  $C = \{1, 2, 3, \dots, m\}$  e  $S = \{S_1, S_2, S_3, \dots, S_p\}$ , tem-se que para todo  $\varepsilon > 0$ , não existe uma  $(1 - \varepsilon) \ln m$ -aproximação para  $(C, S)$ , a menos que  $P=NP$ .*

**Corolário 5.16.** MIN-COLORED- $(s, t)$ -CUT restrito a grafos bipartidos planares não pode ser aproximado por um fator de  $(1 - \varepsilon) \ln(p)$  para qualquer constante  $\epsilon > 0$  a menos que  $P = NP$ .

*Demonstração.* Tome o grafo  $G^c$  construído na Seção 5.3 e para cada um dos  $m$   $(s, t)$ -caminhos  $P_i = (s, p_i^1, p_i^2, \dots, p_i^{d(i)-1}, t)$  execute o seguinte procedimento. Se  $d(i)$  é par, então mantenha  $P_i$  e suas cores inalteradas. Se  $d(i)$  é ímpar, então substitua  $P_i$  por  $P_i^* = (s, p_i^1, p_i^2, \dots, p_i^{d(i)-1}, p_i^{d(i)}, t)$ , tal que  $c(\{p_i^{d(i)-1}, p_i^{d(i)}\}) = c(\{p_i^{d(i)}, t\})$  e as demais arestas de  $P_i^*$  têm as suas cores inalteradas. Desse modo obteremos um  $(s, t)$ -grafo  $H^c$  planar que possui apenas ciclos pares, ou seja,  $H^c$  é um grafo bipartido planar com arestas coloridas. Neste ponto, é fácil ver que  $H^c$  tem um  $(s, t)$ -corte  $[S, T]$  com capacidade  $k$  se e somente se  $G^c$  também possui tal corte. Logo, pelo Teorema 5.2 existe um subconjunto de  $S$  de tamanho  $k$  que cobre  $C$  se e somente se  $H^c$  tem um  $(s, t)$ -corte  $[S, T]$  com capacidade  $k$ .

Como podemos observar, a construção descrita acima é uma S-redução. De fato, o algoritmo de contrução de  $G^c$  a partir de uma instância de SET COVER dado na Seção 5.3 é polinomial e a transformação de  $G^c$  em  $H^c$  é linear. Logo o algoritmo que leva uma instância de SET COVER em  $H^c$  é polinomial. Além disso, como foi mencionado no parágrafo acima, é possível ir de uma solução de  $H^c$  para uma solução de  $G^c$  em tempo linear e a NP-completude de MIN-COLORED- $(s, t)$ -CUT combinada com esse fato garante que existe um algoritmo de tempo polinomial que leva soluções de MIN-COLORED- $(s, t)$ -CUT para  $H^c$  em soluções de SET COVER para  $(C, S)$ . Note que o número de cores do grafo  $G^c$  é igual ao número de elementos da instância de SET COVER. Logo pelo Lema 5.15 e pela Proposição 1.1 temos o requerido.  $\square$

Embora o resultado de inaproximabilidade para o problema MIN-COLORED- $(s, t)$ -CUT resulte diretamente da S-redução de SET COVER para o problema em grafos bipartidos planares, o mesmo não se aplica tão diretamente para grafos completos.

**Corolário 5.17.** MIN-COLORED- $(s, t)$ -CUT em grafos completos não pode ser aproximado por um fator de  $\frac{(1-\varepsilon)}{2} \ln(p)$  para qualquer constante  $\varepsilon > 0$  a menos que  $P = NP$ .

*Demonstração.* Considere a redução apresentada no Corolário 5.5. Pelo Corolário temos que  $A = \text{MIN-COLORED-}(s, t)\text{-CUT}$  com instância  $G''$  possui uma

solução de custo  $k + 1$  se e somente se  $B = \text{setcover}$  com a instância associada possui uma solução de custo  $k$ . Sejam  $OPT_A(G'')$  e  $OPT_B(G')$  soluções ótimas para MIN-COLORED- $(s, t)$ -CUT e para SET COVER, respectivamente. Pela construção apresentada no Corolário 5.5 sabemos que  $|OPT_A(G'')| = |OPT_B(G')| + 1$ . Sendo assim, a partir de uma solução de custo  $(1 - \varepsilon) \ln(c)(|OPT_A(G'')| - 1)$  para MIN-COLORED- $(s, t)$ -CUT obtém-se uma solução de custo  $(1 - \varepsilon) \ln(c)(|OPT_B(G')|)$  para SET COVER, que sabemos não existir a menos que  $P = NP$ . Sendo  $x$  o fator tal que  $x \cdot |OPT_A(G'')| = (1 - \varepsilon) \ln(p)(|OPT_A(G'')| - 1)$  temos:

$$\begin{aligned} x \cdot |OPT_A(G'')| &= (1 - \varepsilon) \ln(p) \cdot |OPT_A(G'')| - (1 - \varepsilon) \ln(p) \Rightarrow \\ &\Rightarrow x = (1 - \varepsilon) \ln(p) - \frac{(1 - \varepsilon) \ln(p)}{|OPT_A(G'')|} \end{aligned}$$

Observe que  $|OPT_A(G'')| \geq 2$ , pois a cor  $p+1$  estará sempre presente em qualquer corte de arestas de  $G''$  além das cores que já existiam em  $G^c$ . Então

$$x \geq (1 - \varepsilon) \ln(p) - \frac{(1 - \varepsilon)}{2} \ln(p) = \frac{(1 - \varepsilon)}{2} \ln(p).$$

□

## 5.6 Resultados polinomiais para o corte colorido mínimo

Para a demonstração do próximo teorema, vamos precisar dos dois lemas a seguir, devidos a Zhang e Fu [71] e que fazem uso de um resultado de Robertson e Seymour [58] e também faremos uso do problema MINIMUM MONOTONE WEIGHTED SAT.

**Lema 5.18.** [71] *Dados dois vértices  $s$  e  $t$  e uma aresta  $e$  em um grafo simples, encontrar um caminho  $P$  que conecta  $s$  a  $t$  e contém a aresta  $e$ , ou verificar que tal caminho não existe, pode ser feito em tempo polinomial.*

**Lema 5.19.** [71] *Dados dois vértices  $s$  e  $t$  e duas arestas  $e_1 = \{u_1, v_1\}$  e  $e_2 = \{u_2, v_2\}$  em um grafo simples, encontrar um caminho  $P$  que conecta  $s$  a  $t$  e contém as arestas  $e_1$  e  $e_2$ , ou verificar que tal caminho não existe, pode ser feito em tempo polinomial.*

A seguir, introduzimos o problema MINIMUM MONOTONE WEIGHTED SAT que será utilizado na demonstração do Teorema 5.20.

MINIMUM MONOTONE WEIGHTED SAT

INSTÂNCIA: Uma terna  $I = (U, C, w)$ , na qual  $U = \{x_1, x_2, \dots, x_n\}$  é um conjunto de variáveis;  $C = \{c_1, c_2, \dots, c_m\}$  é um conjunto de cláusulas no qual cada variável ocorre em no máximo duas cláusulas e nenhuma cláusula contém variáveis negadas;  $w$  uma função peso  $w : V(C) \rightarrow \mathbb{R}$  definida no conjunto  $V(C)$  das variáveis de  $U$  que estão presentes nas cláusulas de  $C$ ; e  $k$  é um inteiro.

PERGUNTA: Existe uma atribuição de valores verdade  $\eta$  para  $U$  com todas as cláusulas satisfeitas, de modo que a soma dos pesos das variáveis verdadeiras de  $V(C)$  é no máximo  $k$ ?

**Teorema 5.20.** MIN-COLORED- $(s, t)$ -CUT pode ser solucionado em tempo polinomial quando toda cor aparece em no máximo dois  $(s, t)$ -caminhos de  $G$ , sendo  $G$  um grafo de entrada com  $(s, t)$ -caminhos de comprimentos quaisquer.

*Demonsitração.* Observe que encontrar, caso exista, um caminho entre  $s$  e  $t$  passando por uma determinada aresta  $e$  pode ser executado em tempo polinomial de acordo com o Lema 5.18. Além disso, um segundo e um terceiro caminhos entre  $s$  e  $t$  passando por  $e$ , caso existam, devem não conter ao menos uma aresta de cada um dos caminhos anteriores que passam por  $e$ . Sendo assim tais caminhos, caso existam, também podem ser obtidos em tempo polinomial. Note também que se a cor  $i$  aparece em no máximo dois  $(s, t)$ -caminhos, então toda aresta colorida com a cor  $i$  ocorre em no máximo dois  $(s, t)$ -caminhos, caso contrário teríamos mais de dois  $(s, t)$ -caminhos apresentando arestas coloridas com a cor  $i$ . Como esse resultado vale para todas as cores da coloração de arestas, podemos concluir que toda aresta do  $(s, t)$ -grafo  $G$  aparece em no máximo dois  $(s, t)$ -caminhos. Esse fato nos permite reconhecer em tempo polinomial se toda cor aparece em no máximo dois  $(s, t)$ -caminhos de  $G$ , assim como enumerar todos os  $(s, t)$ -caminhos de  $G$  em caso positivo, ou seja, o reconhecimento dos grafos que cumprem a hipótese do teorema pode ser feito em tempo polinomial.

Portanto, feita a enumeração de todos os  $(s, t)$ -caminhos de  $G$  em tempo polinomial, podemos em seguida, construir em tempo polinomial uma instância  $F$  de MINIMUM MONOTONE WEIGHTED SAT, na qual cada cláusula representa um  $(s, t)$ -caminho e cada cor representa uma variável, de tal forma que  $F$  possua uma atribuição satisfatóvel de peso  $k$  se e somente se  $G$  possuir um corte de capacidade  $k$ . A seguir ilustramos com o auxílio da Figura 5.9 a construção de  $F$ . Neste ponto, para resolver MIN-COLORED- $(s, t)$ -CUT basta executar em  $F$  o algoritmo polinomial para MINIMUM MONOTONE WEIGHTED SAT (veja o artigo de Porschen e Speckenmeyer [56]).  $\square$

A Figura 5.9 apresenta um  $(s, t)$ -grafo no qual cada cor aparece em no máximo dois  $(s, t)$ -caminhos. A cor 1 aparece nos caminhos  $(s, a, b, c, t)$  e  $(s, a, b, c, e, f, t)$ ;

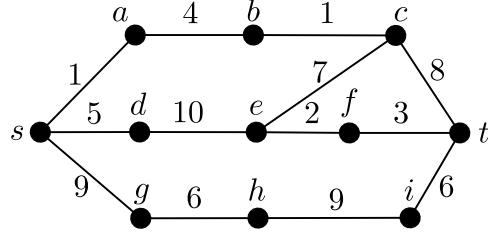


Figura 5.9: Exemplo de um  $(s, t)$ -grafo no qual cada cor aparece em no máximo dois  $(s, t)$ -caminhos.

a cor 2 aparece nos caminhos  $(s, a, b, c, e, f, t)$  e  $(s, d, e, f, t)$ ; a cor 3 aparece em  $(s, a, b, c, e, f, t)$  e  $(s, d, e, f, t)$ ; a cor 4 aparece em  $(s, a, b, c, t)$  e  $(s, a, b, c, e, f, t)$ ; a cor 5 aparece em  $(s, d, e, c, t)$  e  $(s, d, e, f, t)$ ; a cor 6 aparece em  $(s, g, h, i, t)$ ; a cor 7 aparece em  $(s, a, b, c, e, f, t)$  e  $(s, d, e, c, t)$ ; a cor 8 aparece em  $(s, a, b, c, t)$  e  $(s, d, e, c, t)$ ; a cor 9 aparece em  $(s, g, h, i, t)$  e a cor 10 aparece em  $(s, d, e, c, t)$  e  $(s, d, e, f, t)$ . Note que o  $(s, t)$ -grafo dado possui 5  $(s, t)$ -caminhos distintos, a saber:  $(s, a, b, c, t)$ ,  $(s, a, b, c, e, f, t)$ ,  $(s, d, e, c, t)$ ,  $(s, d, e, f, t)$  e  $(s, g, h, i, t)$ . Construimos a instância  $F$  de MINIMUM MONOTONE WEIGHTED SAT fazendo  $U = \{x_1, x_2, \dots, x_{10}\}$ ,  $C = \{c_1, c_2, \dots, c_5\}$ ,  $c_1 = (x_1, x_4, x_8)$ ,  $c_2 = (x_1, x_4, x_7, x_2, x_3)$ ,  $c_3 = (x_5, x_{10}, x_7, x_8)$ ,  $c_4 = (x_5, x_{10}, x_3, x_2)$ ,  $c_5 = (x_9, x_6)$  e atribuindo o peso unitário a cada uma das variáveis de  $U$ . Note que a atribuição que torna verdadeiras apenas as variáveis  $x_1, x_5$  e  $x_9$  torna  $C$  satisfatível com peso 3 e essa atribuição corresponde a um corte de arestas colorido com as cores 1, 5 e 9 (por exemplo, o corte  $[S, V \setminus S]$  com  $S = \{s\}$ ). Do mesmo modo, o corte de arestas  $[S, V \setminus S]$  com  $S = \{s, a, b, c, d, g\}$  apresenta as cores 6, 7, 8, 10 e corresponde à atribuição que torna verdadeiras apenas as variáveis  $x_6, x_7, x_8, x_{10}$  e faz  $C$  satisfatível.

**Corolário 5.21.** MIN-COLORED- $(s, t)$ -CUT pode ser resolvido em tempo polinomial quando cada cor ocorre no máximo em duas arestas e cada  $(s, t)$ -caminho tem comprimento dois.

*Demonstração.* Se cada cor aparece em no máximo duas arestas e cada  $(s, t)$ -caminho tem comprimento dois, isto significa que cada cor poderá aparecer em no máximo dois  $(s, t)$ -caminhos. Logo estamos nas condições da hipótese do Teorema 5.20 e o resultado segue diretamente.  $\square$

A seguir apresentamos um resultado sobre a versão que não faz uso de um par  $(s, t)$  fixo, permitindo solucionar o problema mais geral MIN-COLORED-CUT.

**Lema 5.22.** MIN-COLORED-CUT pode ser resolvido em tempo polinomial quando para todo par  $s, t \in V(G)$  existe um  $(s, t)$ -corte não colorido de tamanho constante, e pode ser resolvido em tempo  $O(p^5 \cdot m)$  em grafos planares.

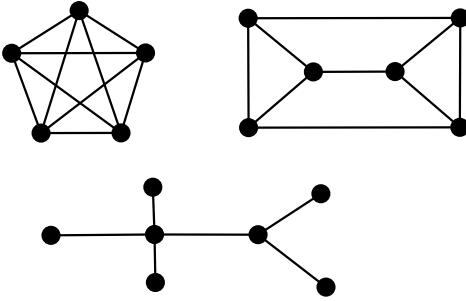


Figura 5.10: Exemplos de grafos tais que para todo par de vértices distintos  $s, t \in V(G)$ , sempre existe um  $(s, t)$ -corte de arestas não colorido de tamanho constante: os grafos com grau máximo limitado,  $k$ -regulares e as árvores fazem parte dessa classe.

*Demonstração.* Dados  $s, t \in V(G)$  quaisquer, se sempre existir um  $(s, t)$ -corte de arestas não colorido de tamanho constante  $q$ , então o corte colorido correspondente a ele tem capacidade no máximo  $q$ , já que o número de cores presentes num corte de arestas é sempre menor ou igual ao número de arestas desse corte. A Figura 5.10 mostra exemplos de grafos tais que para todo par de vértices distintos  $s, t \in V(G)$ , sempre existe um  $(s, t)$ -corte de arestas não colorido de tamanho constante (os grafos  $k$ -regulares fazem parte dessa classe). Isso limita nossas possibilidades de escolhas de cores e torna o problema solucionável em tempo polinomial, pois basta analisarmos as remoções das combinações de até  $q$  cores escolhidas entre as  $p$  cores da coloração de arestas de  $G$ , resultando num algoritmo de tempo  $O(p^q)$ . Para o caso planar, utilizamos a propriedade de que em todo grafo planar existe pelo menos um vértice  $v$  tal que  $d(v) \leq 5$  [8], ou seja, é possível obter um corte de arestas  $[v, V(G) \setminus \{v\}]$  de cardinalidade no máximo 5. A Figura 5.11 ilustra essa situação. Consequentemente, esse corte dá origem a um corte colorido com capacidade  $C \leq 5$  e assim basta testarmos a remoção de até  $C$  cores escolhidas entre as  $p$  cores da coloração de arestas de  $G$ .  $\square$

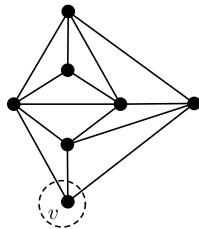


Figura 5.11: Um grafo planar com um vértice  $v$  de grau menor ou igual a 5.

Arestas por cor	$ (s, t)\text{-caminho máximo} $	$\# (s, t)\text{-caminhos por cor}$	Complexidade
3	2	3	NP-completo
2	3	3	NP-completo
—	—	2	Polinomial
2	2	—	Polinomial

Tabela 5.1: Tabela apresentando um resumo dos resultados de complexidade para MIN-COLORED- $(s, t)$ -CUT.

## 5.7 Resultados de NP-completude para o corte colorido máximo

Como sabemos, os problemas MAX-COLORED-CUT e COLORFUL CUT são trivialmente polinomiais para grafos bipartidos, da mesma forma que SIMPLE MAX-CUT, já que é sempre possível encontrar um corte contendo todas as arestas do grafo. Contudo, quando consideramos grafos completos, os problemas MAX-COLORED-CUT e COLORFUL CUT passam a ser NP-completos, diferentemente do que ocorria com SIMPLE MAX-CUT. A seguir provamos esses resultados.

**Teorema 5.23.** MAX-COLORED-CUT permanece NP-completo mesmo quando restrito a grafos completos.

*Demonstração.* Seja  $G$  uma instância de SIMPLE MAX-CUT. Vamos construir uma instância  $G'$  de MAX-COLORED-CUT adicionando a  $G$  um vértice isolado  $v_1$  e atribuindo a cada aresta de  $G$  uma cor exclusiva. Note que  $G$  tem um corte simples de tamanho  $k$  se e somente se  $G'$  tem um corte colorido de capacidade  $k$ . Finalmente, crie arestas conectando todos os pares de vértices não adjacentes em  $G'$  e atribua a todas essas novas arestas uma nova cor  $c'$ , obtendo um grafo completo  $G''$ . A Figura 5.12 ilustra esse procedimento. Como podemos observar, a cor  $c'$  sempre ocorre num corte máximo colorido de  $G''$  e assim  $G''$  possui um corte máximo colorido de tamanho  $k + 1$  se e somente se  $G$  possui um corte máximo de tamanho  $k$ .  $\square$

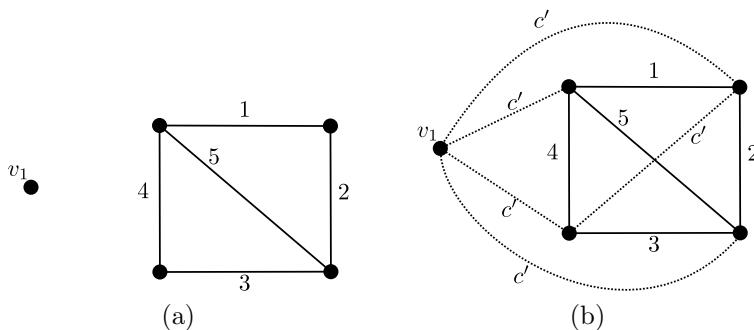


Figura 5.12: (a) Instância  $G'$  de MAX-COLORED-CUT e (b) o grafo completo  $G''$ .

No Teorema 5.24 mostramos que COLORFUL CUT é NP-completo em grafos planares a partir de uma redução de 3-SAT que definimos a seguir.

**3-SAT**

**INSTÂNCIA:** Uma fórmula conjuntiva  $I = (U, C)$ , na qual  $U = \{x_1, x_2, \dots, x_n\}$  é um conjunto de variáveis e  $C = \{c_1, c_2, \dots, c_m\}$  é um conjunto de cláusulas tal que cada cláusula  $c_i \in C$  é formada por 3 literais do conjunto  $U$ .

**PERGUNTA:** Existe uma atribuição de valores verdade  $\eta$  para  $U$  com todas as cláusulas satisfeitas?

**Teorema 5.24.** COLORFUL CUT é NP-completo em grafos planares.

*Demonstração.* Seja  $I = (U, C)$  uma instância de 3-SAT. Vamos construir em tempo polinomial uma instância planar  $G^c = (V, E, f)$  de COLORFUL CUT tal que  $I = (U, C)$  é satisfatível se e somente se  $G^c$  possui um corte  $\partial S$  com todas as cores da coloração.

Para cada cláusula  $c_j = (x, y, z) \in C$  vamos construir uma clique de tamanho 3 na qual cada aresta representa uma das literais de  $c_j$ . Teremos então um grafo  $G'$  formado por  $m$  componentes conexas que são cliques de tamanho 3. A partir de  $G'$  vamos construir o grafo  $G^c$ , que é um multigrafo, associando arestas coloridas a cada um dos literais presentes nas cláusulas de  $C$ , do modo descrito a seguir:

- Para cada par  $\{x_i^j, \bar{x}_i^k\}$  no qual os inteiros  $j$  e  $k$  contabilizam as ocorrências dos literais  $x_i$  e  $\bar{x}_i$  nas cláusulas de  $C$ , respectivamente, atribua uma cor  $S_i^{j,k}$ .
- A aresta que representa o literal  $x_i^j$  em  $G'$  será substituída pelas arestas paralelas coloridas com as cores  $S_i^{j,k}$  em  $G$  para todo  $k$ . Do mesmo modo, a aresta que representa o literal  $\bar{x}_i^k$  em  $G'$  será substituída pelas arestas paralelas coloridas com as cores  $S_i^{j,k}$  para todo  $j$ .

A Figura 5.13 ilustra o grafo  $G'$  associado à instância  $I = (U, C)$  na qual  $U = \{x_1, x_2, x_3\}$  e  $C = \{c_1, c_2, c_3\}$  com  $c_1 = (x_1 \vee \bar{x}_2 \vee \bar{x}_3)$ ,  $c_2 = (\bar{x}_1 \vee x_2 \vee x_3)$  e  $c_3 = (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$  e a Figura 5.14 ilustra o multigrafo  $G$  correspondente com as arestas coloridas.

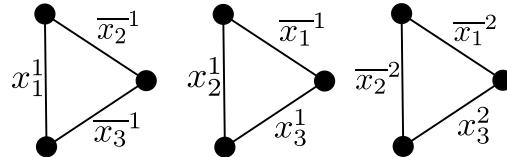


Figura 5.13: Grafo  $G'$  associado à instância  $(x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$ .

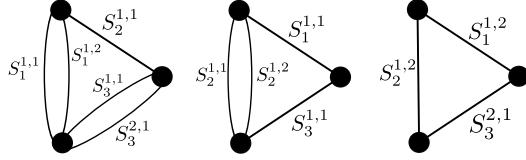


Figura 5.14: Multigrafo  $G$  associado ao grafo  $G'$  com arestas coloridas.

Suponhamos que  $I$  seja satisfatível. Sem perda de generalidade, podemos supor que toda variável aparece pelo menos uma vez positiva e pelo menos uma vez negativa, caso contrário, poderíamos atribuir o mesmo valor verdade a todas as literais presentes e o problema seria trivialmente satisfatível.

Como  $I$  é satisfatível, então toda cláusula de  $C$  possui pelo menos uma literal verdadeira. Caso haja mais de uma literal verdadeira na cláusula, escolha uma delas, digamos  $x_i$ . Então  $\bar{x}_i$  é falsa. Ponha as arestas coloridas correspondentes à variável verdadeira escolhida  $x_i$  em uma das partes da partição e as demais arestas coloridas correspondentes às literais não escolhidas em  $C$ , no corte. Essa construção dá origem a um corte de arestas com todas as cores. De fato, de acordo com a suposição inicial, para cada ocorrência  $x_i^j$  verdadeira, há uma outra de  $\bar{x}_i^k$  falsa, a qual fará com que a cor  $S_i^{j,k}$  correspondente às arestas que representam  $x_i^j$  e  $\bar{x}_i^k$  e está perdida em relação a  $x_j$ , seja recuperada na parte do corte referente à cláusula que apresenta  $\bar{x}_i^k$ . Portanto o corte de arestas construído apresenta todas as cores. A Figura 5.15 ilustra um corte com todas as cores para o multigrafo  $G$  correspondente à instância  $(x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$ , considerando-se a atribuição  $x_1 = V$ ,  $x_2 = V$  e  $x_3 = V$ .

Por outro lado, suponha que  $G^c$  possui um corte que utiliza todas as cores da coloração. Sem perda de generalidade, podemos supor que toda clique possui arestas no corte. De fato, como o corte utiliza todas as cores, se houvesse uma clique inteiramente contida em uma das partes da partição, poderíamos escolher qualquer um de seus vértices e colocá-lo na outra parte, sem prejuízo, pois todas as cores continuariam sendo utilizadas no corte.

A partir desse corte construimos uma atribuição  $\eta$  que satisfaz  $I$  pondo  $x_i$  verdadeiro se as arestas correspondentes a algum literal  $x_i^j$  encontram-se dentro de uma das partes da partição em alguma das  $m$  cliques e  $x_i$  falso caso as arestas correspondentes a algum literal  $\bar{x}_i^k$  encontram-se dentro de uma das partes dessa mesma partição. Note que não existe um par de literais  $\{x_i^j, \bar{x}_i^k\}$  tal que as arestas correspondentes a  $x_i^j$  e a  $\bar{x}_i^k$  estejam, respectivamente, na mesma parte da partição, pois nesse caso a cor  $S_i^{j,k}$  seria perdida e não teríamos um corte com todas as cores presentes. Além disso, como cada clique possui as arestas correspondentes a algum literal dentro de uma das partes da partição, claramente essa atribuição parcial satisfaz todas as cláusulas. Observe que, se após a atribuição parcial realizada, alguma das

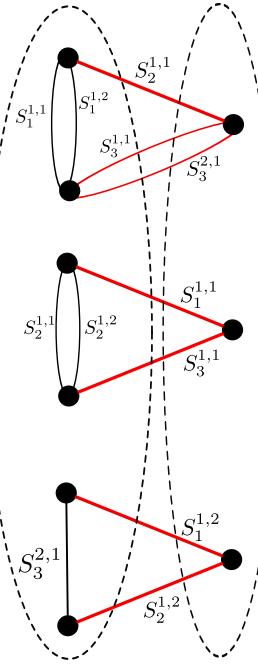


Figura 5.15: Corte de arestas do multigrafo  $G$  associado à instância  $(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$  com atribuição  $x_1 = V$ ,  $x_2 = V$  e  $x_3 = V$ .

variáveis de  $U$  não tiver valor determinado, podemos atribuir a ela qualquer valor, o que não afetará o resultado anterior.

Essa propriedade pode ser estendida para grafos simples, substituindo cada aresta colorida com a cor  $S_i^{j,k}$  por um caminho de comprimento três de modo que a cor da aresta central desse caminho seja  $S_i^{j,k}$  e as arestas das extremidades recebam duas cores novas. A Figura 5.16 ilustra esse procedimento. Não é difícil ver que o multigrafo  $G$  tem um corte colorido que utiliza todas as cores da coloração original se e somente se o grafo simples  $H$  associado tem um corte colorido que utiliza todas as cores da nova coloração. A Figura 5.17 ilustra o corte com todas as cores correspondente ao grafo  $H$ . De fato, se  $G$  tem um corte com todas as cores, então toda cor  $S_i^{j,k}$  está representada no corte e correspondentemente no grafo simples  $H$  podemos dispor alternadamente os vértices do caminho correspondente a  $S_i^{j,k}$  nas partes da partição de modo que as arestas das novas cores criadas estejam sempre presentes no corte. Por outro lado, se o grafo simples  $H$  apresenta um corte de arestas com todas as cores da nova coloração, em particular todas as cores  $S_i^{j,k}$  estão presentes nesse corte. Contraindo as arestas coloridas pelas cores que não são da forma  $S_i^{j,k}$  obteremos exatamente um corte de  $G$  com todas as cores da coloração original.

□

Vamos mostrar no teorema seguinte, que mesmo em subclasses de grafos planares muito particulares o problema do corte colorido máximo permanece NP-completo.

**Teorema 5.25.** COLORFUL CUT é NP-completo mesmo em grafos planares nos

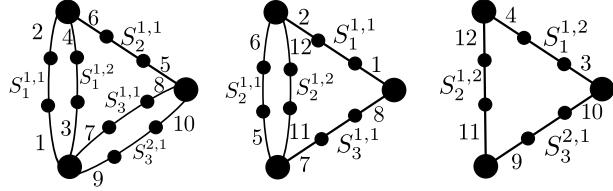


Figura 5.16: Grafo simples  $H$  associado ao multigrafo  $G$ .

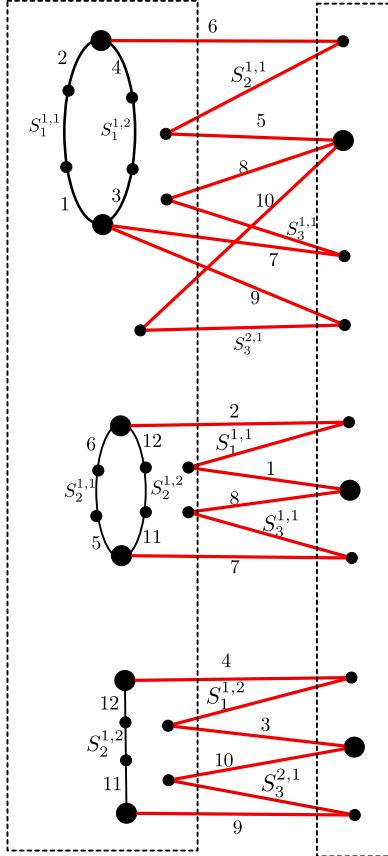


Figura 5.17: Corte de arestas associado ao grafo  $H$ .

quais cada cor ocorre no máximo duas vezes e cada vértice tem grau no máximo 3.

*Demonação.* Considerando o problema 3SAT no qual cada variável aparece no máximo três vezes e cada literal aparece no máximo duas vezes, que sabemos ser NP-completo [37], vamos construir a partir do multigrafo  $G$  do Teorema 5.24, um grafo planar simples  $G''$  em que cada cor ocorre no máximo duas vezes e cada vértice tem grau no máximo 3. Podemos observar que em  $G$  cada cor  $S_i^{j,k}$  aparece exata-mente duas vezes, pois está associada apenas às literais  $x_i^j$  e  $\bar{x}_i^k$ . Para cada par de arestas paralelas de  $G$  com extremos  $u$  e  $v$ , vamos substituir cada um desses extremos por uma aresta, isto é,  $u$  se transforma em  $\{u, u_1\}$  e  $v$  em  $\{v, v_1\}$ , lembrando que esse procedimento deve ser repetido para cada par de arestas paralelas. Com isso, os vértices  $u$  e  $v$  passam a ter ambos grau 2, enquanto  $u_1$  e  $v_1$  possuem ambos grau 3. Vamos colorir as arestas que têm  $u$  como um de seus extremos com a cor

$u$ , do mesmo modo que as arestas que têm  $v$  como extremo receberão a cor  $v$ . Em seguida procedemos como no final do Teorema 5.24, transformando as arestas paralelas em caminhos de comprimento 3, coloridos como no teorema anterior. Desse modo, obteremos um grafo planar  $G''$  com grau máximo 3 no qual cada cor ocorre no máximo duas vezes. A Figura 5.18 ilustra esse procedimento. Podemos observar sem muita dificuldade que  $G''$  tem um corte com todas as cores da coloração resultante se e somente se  $G$  possui um corte com todas as cores da coloração original. De fato, sabemos do Teorema 5.24 que  $G$  possui um corte com todas as cores da coloração original se e somente se  $H$  possui um corte com todas as cores da coloração modificada. Basta mostrar que  $H$  possui um corte com todas as cores da coloração modificada se e somente se  $G''$  possui um corte com todas as cores da coloração final. Observe que se  $H$  possui um corte com todas as cores da coloração modificada, então cada cor  $S_i^{j,k}$  e suas adjacentes no caminho de comprimento 3 sempre comparecem no corte.  $\square$

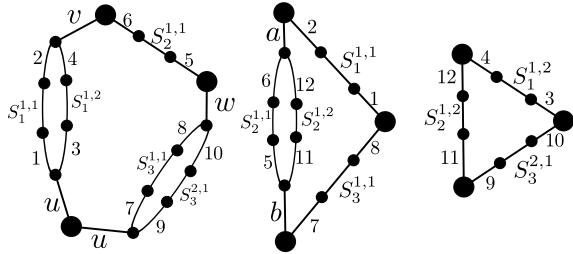


Figura 5.18: Grafo  $G''$ .

O próximo teorema mostra que mesmo quando cada classe de cor induz uma clique, o problema do corte colorido máximo é NP-completo. Para essa demonstração utilizaremos uma redução de NOT ALL EQUAL 3SAT que é NP-completo [61]. Representaremos abreviadamente este problema por NAE 3SAT.

NAE 3SAT

INSTÂNCIA: Uma fórmula conjuntiva  $I = (U, C)$ , na qual  $U = \{u_1, u_2, \dots, u_n\}$  é um conjunto de variáveis e  $C = \{c_1, c_2, \dots, c_m\}$  é um conjunto de cláusulas tal que cada cláusula  $c_i \in C$  é formada por 3 literais do conjunto  $U$ .

PERGUNTA: Existe uma atribuição de valores verdade  $\eta$  para  $U$  de modo que cada cláusula tenha uma literal verdadeira e uma falsa?

**Teorema 5.26.** COLORFUL CUT é NP-completo mesmo quando cada classe de cor induz uma clique.

*Demonstração.* Seja  $I = (U, C)$  uma instância de NAE 3SAT na qual  $U = \{u_1, u_2, \dots, u_n\}$  e  $C = \{c_1, c_2, \dots, c_m\}$ . Vamos construir em tempo polinomial no

tamanho de  $I$ , uma instância  $G^c = (V, E, f)$  de COLORFUL CUT tal que  $I = (U, C)$  é NAE satisfatível se e somente se  $G^c = (V, E, f)$  possui um corte de arestas  $\partial S$  contendo todas as cores da coloração  $f$ .

O grafo aresta-colorido  $G^c = (V, E, f)$  é construído do seguinte modo:

- Para cada cláusula  $c_j = (x, y, z) \in C$  construa uma clique  $\{x_j, y_j, z_j\}$  com todas as arestas coloridas pela cor  $j$ .
- Para cada variável  $u_i \in U$  crie os vértices  $a_i$  e  $b_i$  em  $V$  de modo que  $a_i$  seja unicamente adjacente a todas as ocorrências positivas da variável  $u_i$ , e  $b_i$  seja unicamente adjacente a todas as ocorrências negativas dessa mesma variável.
- Para cada variável  $u_i \in U$  crie uma aresta unindo a primeira ocorrência positiva e a primeira ocorrência negativa dessa variável.
- Excetuando-se as arestas que formam as cliques iniciais, todas as outras arestas devem ser coloridas com cores distintas.

Terminamos dessa forma a construção do grafo. A Figura 5.19 ilustra o grafo associado à instância  $I = (U, C)$  na qual  $U = \{u_1, u_2, u_3\}$  e  $C = \{(u_1 \vee u_2 \vee \bar{u}_3), (\bar{u}_1 \vee \bar{u}_2 \vee \bar{u}_3), (\bar{u}_1 \vee u_2 \vee \bar{u}_3)\}$ .

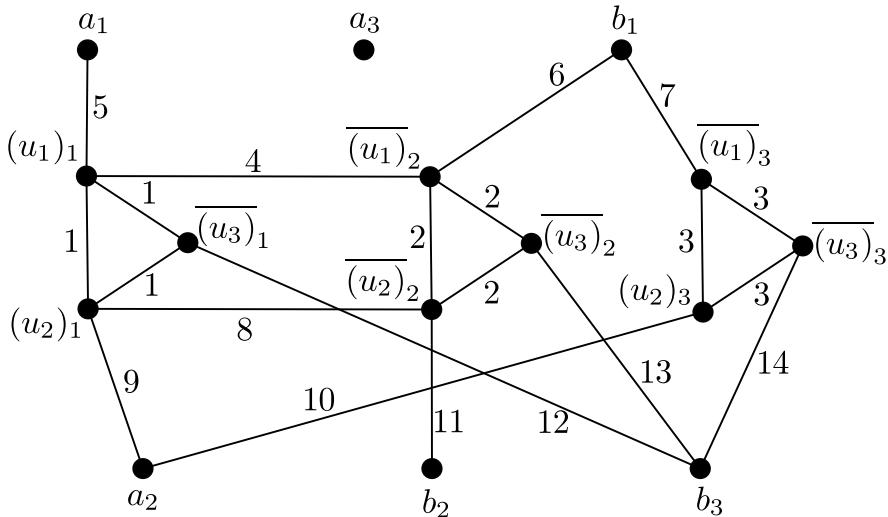


Figura 5.19: Grafo  $G^c$  associado à instância  $I = (U, C)$  na qual  $U = \{u_1, u_2, u_3\}$  e  $C = \{(u_1 \vee u_2 \vee \bar{u}_3), (\bar{u}_1 \vee \bar{u}_2 \vee \bar{u}_3), (\bar{u}_1 \vee u_2 \vee \bar{u}_3)\}$ .

Suponha que  $I = (U, C)$  seja uma instância satisfatível de NAE 3SAT. Seja  $\eta$  uma atribuição de valores verdade para  $U$  que torna a instância satisfatível para NAE 3SAT. O corte  $\partial S$  com todas as cores de  $f$  será obtido da seguinte forma: Se  $u_i$  é verdadeira, então ponha todas as suas ocorrências positivas juntamente com  $b_i$  em  $S$  e ponha todas as suas ocorrências negativas juntamente com  $a_i$  em  $V \setminus S$ . Por construção todas as cores maiores que  $m$  pertencem ao corte  $\partial S$ . Além disso, cada

uma das cores  $j \leq m$  estão presentes no corte já que em cada cláusula há sempre uma ocorrência verdadeira e uma falsa.

Suponha que existe um corte  $\partial S$  contendo todas as cores de  $f$ . Como as cores  $1 \leq j \leq m$  só ocorrem nas arestas das cliques, não é possível que haja alguma clique cujos vértices estejam todos na mesma parte da partição, ou seja, para cada clique há pelo menos um vértice de cada lado, garantindo a presença da cor no corte. Vamos produzir uma atribuição de valores verdade  $\eta$  que torna  $I = (U, C)$  NAE 3SAT satisfatível do seguinte modo: atribuímos valor verdadeiro aos literais correspondentes aos vértices da clique  $\{x_j, y_j, z_j\}$  que pertencem a  $S$ , e valor falso aos literais correspondentes aos vértices da clique que pertencem a  $V \setminus S$ . Observe que essa atribuição é consistente, pois a aresta que une as primeiras ocorrências positiva e negativa da variável  $u_i$  tem uma cor única presente no corte, significando que essas literais possuem valores verdade opostos. Além disso, como todas as ocorrências positivas da variável  $u_i$  são adjacentes ao vértice  $a_i$  por arestas com cores exclusivas, todas presentes no corte, isto significa que todas as ocorrências positivas da variável  $u_i$  estão na mesma parte da partição e portanto recebem o mesmo valor verdade. Um raciocínio análogo garante que todas as ocorrências negativas da variável  $u_i$  estão na mesma parte da partição e recebem todas o mesmo valor verdade.

Observe que no grafo construído cada classe de cor induz uma clique, pois as cores  $1 \leq j \leq m$  só aparecem nas cliques correspondentes às  $m$  cláusulas, enquanto todas as outras arestas possuem cores exclusivas.  $\square$

Para a conveniência do leitor, apresentamos na Figura 5.20 o corte com todas as 14 cores obtido de  $G^c = (V, E, f)$  proveniente da instância de NAE 3SAT dada por  $I = (U, C)$  na qual  $U = \{u_1, u_2, u_3\}$  e  $C = \{(u_1 \vee u_2 \vee \bar{u}_3), (\bar{u}_1 \vee \bar{u}_2 \vee u_3), (\bar{u}_1 \vee u_2 \vee \bar{u}_3)\}$ , satisfazendo a atribuição  $\bar{u}_1 = u_2 = u_3 = V$ .

Já havíamos visto no Teorema 5.23 que MAX-COLORED-CUT é NP-completo mesmo restrito a grafos completos. No próximo teorema vamos provar que o mesmo ocorre com COLORFUL CUT.

**Teorema 5.27.** COLORFUL CUT é NP-completo mesmo quando restrito a grafos completos.

*Demonstração.* Com base na construção feita no Teorema 5.26 vamos mostrar que COLORFUL CUT é NP-completo mesmo quando restrito a grafos completos. Sejam  $I = (U, C)$  uma instância de NAE 3SAT e  $G^c = (V, E, f)$  a instância de COLORFUL CUT produzida no Teorema 5.26 a partir de  $I$ . Construa o grafo aresta-colorido completo  $H^g = (V, E', g)$  a partir de  $G^c$  adicionando arestas a todos os pares de vértices dois a dois não adjacentes, mantendo a coloração original de  $G^c$  para as arestas que já existiam ( $c$  cores) e atribuindo uma nova cor  $c + 1$  a todas as arestas adicionadas. Vamos mostrar que  $I = (U, C)$  é NAE 3SAT satisfatível se e somente

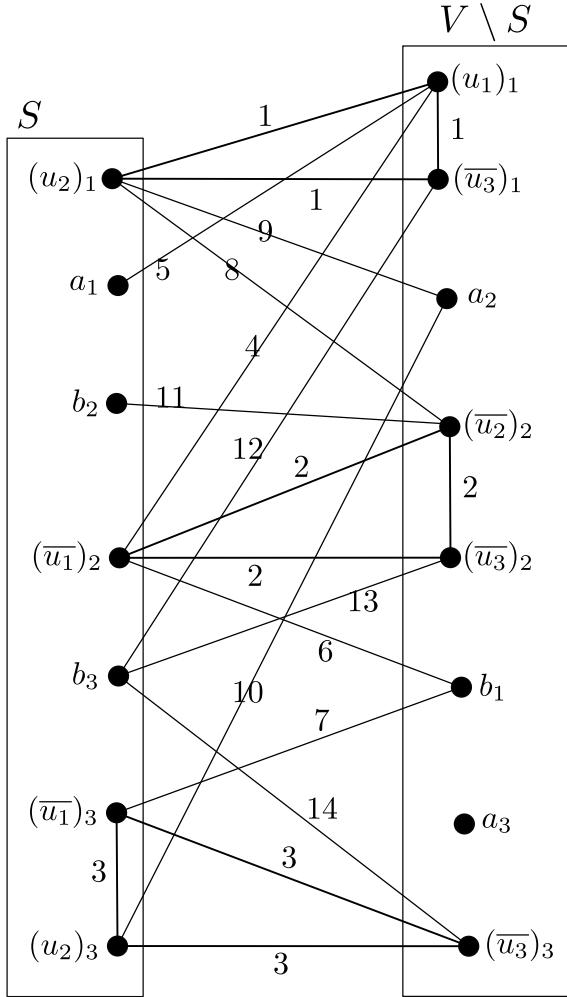


Figura 5.20: Corte com 14 cores obtido de  $G^c = (V, E, f)$  da Figura 5.19 para a atribuição satisfatível  $\bar{u}_1 = u_2 = u_3 = V$ .

se existe um corte de arestas de  $H^g$  com todas as cores da coloração  $g$ , isto é,  $c + 1$  cores.

Pelo Teorema 5.26, temos que  $I = (U, C)$  é NAE 3SAT satisfatível se e somente se existe um corte de arestas de  $G^c$  com todas as cores da coloração  $f$ . Vamos mostrar que existe um corte de arestas de  $G^c$  com todas as cores de  $f$  se e somente se existe um corte de arestas de  $H^g$  com todas as cores da coloração  $g$  e isso nos dará o resultado final desejado.

Suponha que  $G^c$  possui um corte de arestas  $\partial S$  contendo todas as  $c$  cores de  $f$ . Então  $\partial S$  é um corte de arestas de  $H^g$  com  $c + 1$  cores. De fato, como os vértices  $a_i$  e  $b_i$  são não adjacentes em  $G^c$  e se encontram em partes distintas da partição  $[S, V \setminus S]$ , então para essa mesma partição, a aresta  $\{a_i, b_i\}$  de  $H^g$  participará do corte de arestas correspondente nesse grafo, estando colorida com a nova cor  $c + 1$ .

Suponha que  $H^g$  possui um corte de arestas com  $c + 1$  cores. Contraindo todas as arestas coloridas com a nova cor  $c + 1$  recuperaremos o grafo  $G^c$  com a coloração  $f$ . Logo a mesma partição dá origem a um corte de arestas de  $G^c$  com todas as  $c$

cores da coloração  $f$ . □

Para a demonstração do próximo corolário precisaremos de uma nova definição. Dado um grafo  $G = (V, E)$ , dizemos que um conjunto  $O \subset V(G)$  é *odd cycle transversal* se o subgrafo induzido  $G[V \setminus O]$  é bipartido, ou seja, se a remoção dos vértices de  $O$  dá origem a um grafo bipartido. O número mínimo de vértices de  $V(G)$  cuja remoção origina um grafo bipartido é chamado de *odd cycle transversal number* e será representado por  $oct(G)$ .

**Corolário 5.28.** COLORFUL CUT é NP-completo, mesmo quando restrito a grafos planares com odd cycle transversal number no máximo 1.

*Demonstração.* Se  $oct(G) = 0$ , então  $G$  é bipartido e o problema é polinomial. Para mostrar que COLORFUL CUT é NP-completo quando  $oct(G) = 1$  fazemos uma redução de 3-SAT. Seja  $H$  o grafo obtido no Teorema 5.24. Vamos escolher em cada componente conexa de  $H$  um vértice que pertencia originalmente ao multigrafo  $G$  de modo que todos os vértices escolhidos estejam na mesma parte da partição  $(S, V \setminus S)$ . Isso sempre pode ser feito, pois a construção da partição utilizada no Teorema 5.24 considera sempre como arestas não presentes no corte aquelas correspondentes a uma literal verdadeira, o que permite inverter as posições dos vértices pertencentes a uma mesma componente conexa no corte, sem prejuízo. Em seguida, vamos identificar todos os vértices escolhidos como um único vértice, dando origem ao grafo  $F$ , como ilustra a Figura 5.21. Observe que cada um dos vértices escolhidos pertence a todos os ciclos ímpares de sua respectiva componente conexa. Portanto o vértice de  $F$  obtido após a fusão está em todos os ciclos ímpares desse grafo, ou seja,  $oct(F) = 1$ . Após a fusão dos vértices, as arestas presentes no corte de  $H$  se mantêm para  $F$  e assim, pelo Teorema 5.24 temos que COLORFUL CUT é NP-completo quando odd cycle transversal number é 1. □

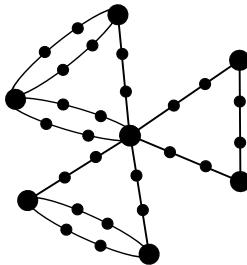


Figura 5.21: Grafo  $F$  obtido após a fusão dos vértices do multigrafo  $G$ .

## 5.8 Algoritmo 2-aproximativo para o corte colorido máximo

**Teorema 5.29.** MAX-COLORED-CUT admite um algoritmo polinomial 2-aproximativo.

*Demonstração.* Dado um grafo aresta-colorido  $G^c = (V, E, c)$  com uma coloração de arestas  $c : E(G) \rightarrow \{1, 2, \dots, p\}$ , sabemos que o número  $p$  de cores não ultrapassa o número  $m$  de arestas, ou seja,  $m \geq p$ . Usando o método probabilístico, Erdős provou que, dado um grafo  $G$  sempre é possível determinar um corte de arestas de  $G$  com metade de suas arestas. Desse modo existe um corte  $\partial S$  tal que  $|\partial S| \geq \frac{m}{2} \geq \frac{p}{2}$ . No caso de uma coloração de arestas própria, o método de Erdős já garante a existência de um corte de arestas que apresenta pelo menos metade das cores dessa coloração, já que a capacidade do corte será dada exatamente pelo número de arestas presentes no corte, para essa situação.

Dado um grafo aresta-colorido  $G^c$ , podemos construir um algoritmo 2-aproximativo para a determinação de um corte de arestas de  $G^c$  com capacidade  $\frac{p}{2}$  usando um argumento semelhante ao do algoritmo guloso apresentado na Seção 1.4 do Capítulo 1. O algoritmo consiste dos seguintes passos:

1. Tome uma ordenação  $(v_1, \dots, v_n)$  dos vértices de  $G^c$ ;
2. Inicie com os subconjuntos  $A = \{v_1\}$  e  $B = \{v_2\}$ ;
3. Para  $i = 3, \dots, n$  faça: se  $C([v_i \cup A, B]) \geq C([A, v_i \cup B])$ , então ponha  $v_i$  em  $A$ ; caso contrário, ponha  $v_i$  em  $B$ .

□

## 5.9 Resultados de complexidade parametrizada para o corte colorido máximo

**Teorema 5.30.** MAX-COLORED-CUT admite um núcleo cúbico quando parametrizado pelo número de cores.

*Demonstração.* Consideremos MAX-COLORED-CUT parametrizado pelo número de cores presentes no corte, isto é  $(G, c, p)$  em que  $G$  é um grafo aresta-colorido com coloração  $c$  e o parâmetro  $p$  é o número de cores presentes no corte. Olhamos para o problema por uma perspectiva diferente: procuramos um subgrafo bipartido contendo o maior número de cores. Observe que, se para alguma cor  $i$ , o conjunto  $E_i$  formado pelas arestas coloridas com a cor  $i$  tiver tamanho maior que  $2p^2 \geq 2\binom{p-1}{2}$ ,

certamente essa cor sempre poderá fazer parte de uma solução. De fato, supondo que já existe um corte com  $p - 1$  cores, devemos ter pelo menos  $p - 1$  arestas de cores distintas nesse corte. Desse modo, deve haver no máximo  $p - 1$  vértices em cada parte dessa partição  $(S, T)$  como extremos destas  $p - 1$  arestas de cores distintas. Se não puder ser acrescentada nenhuma aresta da cor  $i$  a esse corte, significa que todas as arestas coloridas com essa cor possuem ambos os extremos ou em  $S$ , ou em  $T$ . Consequentemente há, no máximo,  $2\binom{p-1}{2} = (p-1)(p-2)$  arestas dessa cor. Portanto, aplicamos a seguinte regra de redução simples: se alguma cor  $i$  satisfaz  $|E_i| > 2p^2$ , então substitua  $(G, c, k)$  por  $(G - E_i, p - 1, k - 1)$ . No pior caso, esse procedimento deverá ser realizado para todas as cores da coloração, resultando em uma instância equivalente com  $O(p^3)$  vértices.  $\square$

**Corolário 5.31.** MAX-COLORED-CUT admite um núcleo cúbico quando parametrizado pelo custo da solução.

*Demonstração.* Seja  $k$  o custo do corte desejado, isto é, o número de cores presentes nesse corte. Usamos a redução descrita na demonstração do Lema 5.30, juntamente com a propriedade bem conhecida de que qualquer grafo com pelo menos  $2p$  arestas contém um subgrafo bipartido com pelo menos  $p$  arestas (veja [25]). Ou seja, se  $p \geq 2k$ , então tomando uma aresta de cada classe de cor, podemos obter um corte com pelo menos  $k$  cores. Portanto, podemos assumir que  $p \leq 2k$ , e consequentemente obtemos um núcleo com  $O(k^3)$  vértices.  $\square$

## 5.10 Resultados polinomiais para o corte colorido máximo

Como uma consequência imediata do Teorema 5.30 temos o seguinte:

**Corolário 5.32.** MAX-COLORED-CUT pode ser resolvido em tempo polinomial em grafos  $G$  coloridos com um número constante de cores.

*Demonstração.* Como vimos no Teorema 5.30, MAX-COLORED-CUT é FPT quando o parâmetro é o número de cores. Logo é solúvel em tempo  $f(c) \cdot n^{O(1)}$ . Desse modo, se  $c$  é constante temos uma solução de tempo  $n^{O(1)}$ , como queríamos.  $\square$

Todos os resultados desse capítulo encontram-se nos artigos “*On Colored Edge Cuts in Graphs*” [49], apresentado no Primeiro Encontro de Teoria da Computação (I ETC), em julho de 2016 e *Sobre Cortes em Grafos Aresta-Coloridos* [31] apresentado no XLVIII SBPO, em novembro de 2016. Em ambos os artigos houve a colaboração dos pesquisadores Ignasi Sau, pesquisador júnior do Laboratoire d’Informatique, Robotique et Microélectronique da Universidade de Montpellier e Uéverton dos Santos

Souza, professor adjunto do Instituto de Computação da Universidade Federal Fluminense.

# Capítulo 6

## Conclusões

Nesse último capítulo apresentamos uma visão geral de todo o trabalho realizado até o presente momento.

No Capítulo 1 estabelecemos o problema MAX CUT, bem como todas as definições preliminares para seu entendimento. Apresentamos também a redução original feita por Karp [48] mostrando que MAX CUT é NP-completo. A partir daí, nos concentramos no Problema SIMPLE MAX CUT, o qual é uma versão simplificada do Problema MAX CUT. Citamos os resultados obtidos para esse problema em termos de complexidade, bem como os artigos referentes a algoritmos de tempo polinomial que resolvem o problema em certas classes de grafos, como por exemplo, grafos planares. Mencionamos também o algoritmo aproximativo de Goemans e Williamson [42] que usa programação semidefinida para obter uma solução aproximada de razão 0,878. Esse algoritmo aproximativo é o melhor algoritmo conhecido até o momento. Sobre o problema geral, baseado no limite de Erdős para SIMPLE MAX CUT, temos uma sugestão que consideramos muito interessante da professora Celina Miraglia Herrera de Figueiredo: analisar o status de complexidade do problema de determinar se um grafo  $G = (V, E)$  dado tem um corte de arestas com exatamente  $\frac{m}{2}$  arestas.

No Capítulo 2 mostramos que SIMPLE MAX CUT considerado na classe dos grafos fortemente cordais é um problema NP-completo, utilizando a redução feita por Bodlaender e Jansen [6] e verificando que o grafo construído por eles é fortemente cordal, através da apresentação de um esquema de eliminação forte para esse grafo. Os resultados do Capítulo 2 foram apresentados no artigo “*A Complexidade do Problema do Corte Máximo para Grafos Fortemente Cordais*”, no XLV SBPO, 2013, e publicado nos Anais do XLV SBPO, 2013 [62]. A respeito de problemas em aberto para o Capítulo 2, um problema que estamos trabalhando consiste em considerar a análise de complexidade de SIMPLE MAX CUT para outras subclasses de grafos cordais como por exemplo a classe dos grafos de intervalo próprios, isto é, os grafos de intervalo que possuem um modelo de interseção no qual nenhum intervalo está propriamente contido em outro.

No Capítulo 3 apresentamos algoritmos aproximativos polinomiais para resolver SIMPLE MAX CUT em grafos split. Inicialmente, utilizando uma cota superior para o tamanho do corte máximo, construímos um algoritmo aproximativo polinomial de razão  $\frac{2}{3}$ . Exibimos um exemplo no qual o corte obtido pela aplicação do algoritmo fornece um corte muito próximo do corte ótimo, sendo  $\frac{7}{8}$  do tamanho do corte ótimo. Esse algoritmo tem a vantagem de determinar, em alguns casos particulares, exatamente o corte máximo. Em seguida, definimos subclasses especiais dos grafos *split*, tais como grafos *split* cheios simples e os *split* cheios múltiplos, apresentando algoritmos que fornecem exatamente um corte máximo para esses grafos. Os resultados desse capítulo encontram-se no artigo “*Approximation Algorithms for Simple Max Cut of Split Graphs*” [63] que foi um dos 5 trabalhos selecionados para concorrer ao prêmio Roberto Diéguez Galvão no XLVI SBPO, 2014 e encontra-se publicado nos Anais do XLVI SBPO, 2014.

No Capítulo 4 analisamos um problema que apresenta SIMPLE MAX CUT como um de seus casos particulares, o problema SIGNED MAX CUT ATLB: dado um grafo sinalizado  $G$ , determinar o maior subgrafo balanceado de  $G$ , isto é, um subgrafo balanceado de  $G$  com o maior número de arestas possível. Esse problema foi estudado, do ponto de vista da complexidade parametrizada, por Crowston, Gutin, Jones e Muciaccia [15]. Através da aplicação de um conjunto de regras de redução, Crowston, Gutin, Jones e Muciaccia demonstraram que, partindo de uma instância  $(G, k)$  do problema é possível determinar uma nova instância  $(G', k')$  chamada *núcleo* de tamanho menor que o de  $(G, k)$  cumprindo as seguintes condições: se  $(G', k')$  responde positivamente ao problema de decisão associado, então  $(G, k)$  também o faz; ou a ordem de  $G'$  é limitada superiormente por uma função do parâmetro  $k$ . Tomando por base estes resultados, mostramos que o problema SIGNED MAX CUT ATLB em grafos  $(r, \ell)$  tem *núcleo* quadrático. Definimos ainda a subclasse dos grafos  $d^*$ -split e provamos que nessa classe o problema tem *núcleo* linear, utilizando para isso um algoritmo de tempo polinomial. Os resultados desse capítulo encontram-se no artigo “*Improved Kernels for Signed Max Cut Parameterized Above Lower Bound on  $(r, \ell)$ -Graphs*” [30] que teve também a colaboração do pesquisador Ignasi Sau. Esse artigo foi submetido ao jornal *Discrete Mathematics and Theoretical Computer Science* em 2016. Cabe aqui ressaltar que em dezembro de 2016 foi apresentado no ISAAC 2016 o artigo “*Linear Kernels and Linear Time Algorithms for Finding Large Cuts*” de Michael Etscheid e Matthias Mnich [28] no qual é determinado um núcleo linear, melhorando consideravelmente tanto os resultados de Crowston, Gutin, Jones e Muciaccia [15], bem como os nossos resultados.

No Capítulo 5 estudamos uma nova versão dos problemas de cortes de arestas máximos e mínimos, introduzindo no grafo considerado uma coloração de arestas e buscando cortes de arestas que apresentam o menor número de cores (MIN-

COLORED-CUT), bem como cortes de arestas com o maior número de cores possível (MAX-COLORED-CUT) ou cortes que utilizam todas as cores da coloração dada (COLORFUL CUT). Consideramos a versão simplificada de MIN-COLORED-CUT na qual são fixados dois vértices  $s, t \in G$  e procura-se um corte de arestas  $[S, T]$  com o mínimo de cores, de modo que  $s \in S$  e  $t \in T$  (MIN-COLORED- $(s, t)$ -CUT). Verificamos que mesmo nesse caso simplificado o problema é NP-completo, em contraponto com o problema clássico do Corte Mínimo Simples(MIN CUT) que é polinomial. Provamos que o problema MIN-COLORED- $(s, t)$ -CUT é NP-completo para grafos bipartidos planares. Ainda que seja considerada a versão de MIN-COLORED- $(s, t)$ -CUT parametrizado pelo número de cores da solução dada, mostramos que esse problema é  $W[2]$ -difícil. Mostramos que o mesmo ocorre em grafos completos: MIN-COLORED- $(s, t)$ -CUT é NP-completo e  $W[2]$ -difícil quando parametrizado pelo número de cores da solução procurada. Por outro lado, mostramos que tanto MIN-COLORED- $(s, t)$ -CUT como MIN-COLORED-CUT são FPT quando parametrizados pelo número de cores da coloração de arestas, o que permite resolver facilmente esses problemas quando o número de cores da coloração de arestas é fixo. Encontramos também situações em que MIN-COLORED- $(s, t)$ -CUT e MIN-COLORED-CUT são polinomiais: o primeiro problema é polinomial para  $(s, t)$ -grafos nos quais cada cor aparece em no máximo dois  $(s, t)$ -caminhos de comprimentos quaisquer, e o segundo é polinomial se dados dois vértices quaisquer  $s, t \in V$ , sempre existe um  $(s, t)$ -corte simples de tamanho constante. No estudo de cortes coloridos máximos, mostramos que MAX-COLORED-CUT é NP-completo em grafos completos e COLORFUL CUT é NP-completo em grafos planares, mesmo quando cada cor aparece em apenas duas arestas e o grau máximo dos vértices é três. Provamos também que COLORFUL CUT é NP-completo quando cada classe de cor induz uma clique. Determinamos um algoritmo polinomial 2-aproximativo para a obtenção de um corte colorido máximo, baseados em um raciocínio análogo ao empregado por Erdős na obtenção de um corte máximo simples. Do ponto de vista parametrizado, mostramos que MAX-COLORED-CUT é FPT e admite um núcleo cúbico, tanto quando parametrizado pelo número de cores da coloração, quanto quando parametrizado pelo número de cores presentes na solução. Observamos, por último, que MAX-COLORED-CUT pode ser resolvido em tempo polinomial quando o número de cores da coloração é constante. Os resultados do Capítulo 6 encontram-se em dois artigos que foram produzidos com a colaboração dos pesquisadores Ignasi Sau e Uéverton Souza: “*On Colored Edge Cuts in Graphs*” [49], apresentado no Primeiro Encontro de Teoria da Computação (I ETC), em julho de 2016 e “*Sobre Cortes em Grafos Aresta-Coloridos*” [31] apresentado no XLVIII SBPO, em novembro de 2016.

Sobre o Capítulo 5 ressaltamos que já foi iniciado um trabalho com os pesquisadores Ignasi Sau e Uéverton Souza sobre o problema MAX-COLORED-CUT com o

Classe	Problemas													
	MAXCUT								MINCUT					
	SIMPLE MAXCUT				EDGE COLORED				SIMPLE MINCUT		EDGE COLORED			
	simples		sinal. ATLB		simples		colorful		simples		simples		$(s, t)$	
C.C.	C.C.	C.P.	C.C.	C.P.	C.C.	C.P.	C.C.	C.P.	C.C.	C.C.	C.C.	C.P.	C.C.	C.P.
geral	NPc [GJS,74]	FPT [CJM,12]	NPc [CGJM,13]	FPT [CGJM,13]	NPc  c	FPT  c	NPc  c	FPT  c	P [EK,72]	O	FPT  c	NPc  c	FPT  c	
split	NPc [BJ,00]	FPT	NPc	FPT	NPc	FPT	O	FPT	P	O	FPT	O	FPT	
$(k, n)$ -split cheios	P	FPT	O	FPT	O	FPT	O	FPT	P	O	FPT	O	FPT	
$d^*$ -split	NPc	FPT	NPc	FPT	O	FPT	O	FPT	P	O	FPT	O	FPT	
cordais	NPc [BJ,00]	FPT	NPc	FPT	O	FPT	O	FPT	P	O	FPT	O	FPT	
fortemente cordais	NPc	FPT	NPc	FPT	O	FPT	O	FPT	P	O	FPT	O	FPT	
$(r, l)$	NPc	FPT	NPc	FPT	O	FPT	O	FPT	P	O	FPT	O	FPT	
completos	P	FPT	O	FPT	NPc	FPT	NPc	FPT	P	O	FPT	NPc	W[2]h  sol.	
planares	P [Had,75]	FPT	O	FPT	NPc	FPT	NPc	FPT	P	P	W[2]h	NPc	W[2]h	
bipartidos planares	P	FPT	O	FPT	NPc	FPT	P	FPT	P	P	W[2]h	NPc	W[2]h  sol.	
OCT $\leq 1$	P	FPT	O	FPT	NPc	FPT	NPc	FPT	P	O	FPT	O	FPT	
$ apc  \leq 3$ $ (s, t) =2$	—	—	—	—	—	—	—	—	—	—	—	NPc	FPT	
$ apc  \leq 2$ $ (s, t)  \leq 3$	—	—	—	—	—	—	—	—	—	—	—	NPc	FPT	
$ c  = \text{const.}$	—	—	—	—	P	FPT	P	FPT	—	P	FPT	P	FPT	
clas.=clique	—	—	—	—	NPc	FPT	NPc	FPT	—	O	FPT	O	FPT	

Tabela 6.1: C.C.=“complexidade clássica”; C.P.=“complexidade parametrizada”; O=“aberto”; P=“polinomial”.

objetivo de determinar a soleira P versus NP-completo quando o *vertex cover number* do grafo é fixado. Já mostramos que estão em P os casos em que o *vertex cover number* é 1, ou 2, e estamos analisando o caso quando *vertex cover number* é maior ou igual a 3. Pensamos que este último trabalho possa também ser estendido para o problema COLORFUL CUT, bem como para a análise de Complexidade Parametrizada em ambos os casos. Pensamos também no problema MIN-COLORED-CUT em grafos aresta-coloridos que é mencionado no artigo de Coudert, Datta, Pérennes, Rivano e Voge [11] e aberto até hoje na literatura. Pensamos na dicotomia P versus NPc de MIN-COLORED-CUT e COLORFUL CUT para grafos  $(r, \ell)$  que ainda não foi estudada, e é claro também pensamos no estudo da complexidade parametrizada desses problemas nessa classe de grafos.

A Tabela 6.1 mostra um resumo de todos os resultados obtidos nessa Tese.

# Referências Bibliográficas

- [1] F. Barahona. Balancing signed graphs of fixed genus in polynomial time. *Depto. de Matematicas, Universidad de Chile, Santiago, Chile*, 1981.
- [2] F. Barahona, M. Grötschel, M. Jünger, and G. Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36(3):493–513, 1988.
- [3] G. Blin, P. Bonizzoni, R. Dondi, R. Rizzi, and F. Sikora. Complexity insights of the minimum duplication problem. *Theoretical Computer Science*, 530:66–79, 2014.
- [4] H. L. Bodlaender, C. M. H. De Figueiredo, M. Gutierrez, T. Kloks, and R. Niedermeier. Simple Max-Cut for Split-Indifference Graphs and Graphs with Few  $P_4$ ’s. In *Experimental and Efficient Algorithms*, pages 87–99. Springer, 2004.
- [5] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *Journal of Computer and System Sciences*, 75(8):423–434, 2009.
- [6] H. L. Bodlaender and K. Jansen. On the complexity of the maximum cut problem. *Nordic Journal of Computing*, 7(1):14–31, 2000.
- [7] B. Bollobás and A. D. Scott. Better bounds for max cut. *Contemporary combinatorics*, 10:185–246, 2002.
- [8] J. A. Bondy and U. S. R. Murty. *Graph theory with applications*, volume 290. Citeseer, 1976.
- [9] A. Brandstädt. Partitions of graphs into one or two independent sets and cliques. *Discrete Mathematics*, 152(1):47–54, 1996.
- [10] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

- [11] D. Coudert, P. Datta, S. Perennes, H. Rivano, and M. E. Voge. Shared risk resource group complexity and approximability issues. *Parallel Process. Lett.*, 17(2):169–184, 2007.
- [12] P. Crescenzi. A short guide to approximation preserving reductions. In *Computational Complexity, 1997. Proceedings., Twelfth Annual IEEE Conference on (Formerly: Structure in Complexity Theory Conference)*, pages 262–273. IEEE, 1997.
- [13] P. Crescenzi, V. Kann, R. Silvestri, and L. Trevisan. Structure in approximation classes. In *International Computing and Combinatorics Conference*, pages 539–548. Springer, 1995.
- [14] R. Crowston, M. Fellows, G. Gutin, M. Jones, F. Rosamond, S. Thomassé, and A. Yeo. Simultaneously satisfying linear equations over f2: Maxlin2 and max-r-lin2 parameterized above average. In *31st International Conference on Foundations of Software Technology and Theoretical Computer Science*, page 229. Citeseer, 2011.
- [15] R. Crowston, G. Gutin, M. Jones, and G. Muciaccia. Maximum balanced subgraph problem parameterized above lower bound. *Theoretical Computer Science*, 513:53–64, 2013.
- [16] R. Crowston, M. Jones, and M. Mnich. Max-cut parameterized above the edwards-erdős bound. In *ICALP*, pages 242–253. Springer, 2012.
- [17] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [18] I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing, STOC ’14*, pages 624–633, New York, NY, USA, 2014. ACM.
- [19] G. A. Dirac. On rigid circuit graphs. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 25, pages 71–76. Springer, 1961.
- [20] R. G. Downey and M. R. Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013.
- [21] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

- [22] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972.
- [23] C. S. Edwards. Some extremal properties of bipartite subgraphs. *Canad. J. Math.*, 25(3):475–483, 1973.
- [24] C. S. Edwards. An improved lower bound for the number of edges in a largest bipartite subgraph. In *Proc. 2nd Czechoslovak Symposium on Graph Theory, Prague*, pages 167–181, 1975.
- [25] P. Erdős. On some extremal problems in graph theory. *Israel Journal of Mathematics*, 3:113–116, 1965.
- [26] P. Erdős. On bipartite subgraphs of graphs. *Mat. Lapok*, 18:283–288, 1967.
- [27] P. Erdős, A. Gyárfás, and Y. Kohayakawa. The size of the largest bipartite subgraphs. *Discrete Mathematics*, 177(1):267–271, 1997.
- [28] M. Etscheid and M. Mnich. Linear Kernels and Linear-Time Algorithms for Finding Large Cuts. In Seok-Hee Hong, editor, *27th International Symposium on Algorithms and Computation (ISAAC 2016)*, volume 64 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:13, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [29] M. Farber. Characterizations of strongly chordal graphs. *Discrete Mathematics*, 43(2):173–189, 1983.
- [30] L. Faria, S. Klein, I. Sau, and R. Sucupira. Improved kernels for signed max cut parameterized above lower bound on  $(r,l)$ -graphs. *arXiv preprint arXiv:1512.05223*, 2015.
- [31] L. Faria, R. Sucupira, S. Klein, I. Sau, and U. Souza. Sobre cortes em grafos aresta-coloridos. XLVIII Simpósio Brasileiro de Pesquisa Operacional, setembro 2016.
- [32] T. Feder, P. Hell, S. Klein, and R. Motwani. List partitions. *SIAM Journal on Discrete Mathematics*, 16(3):449–478, 2003.
- [33] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. Springer, 2006.
- [34] J. Flum and M. Grohe. Parameterized complexity theory, volume xiv of texts in theoretical computer science. an eatcs series, 2006.

- [35] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8(3):399–404, 1956.
- [36] D. Fulkerson and O. Gross. Incidence matrices and interval graphs. *Pacific journal of mathematics*, 15(3):835–855, 1965.
- [37] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, San Francisco, 1979.
- [38] M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-completeness, 1979.
- [39] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63. ACM, 1974.
- [40] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical computer science*, 1(3):237–267, 1976.
- [41] F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B*, 16(1):47–56, 1974.
- [42] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.
- [43] M. Grötschel and G. L. Nemhauser. A polynomial algorithm for the max-cut problem on graphs without long odd cycles. *Mathematical Programming*, 29(1):28–40, 1984.
- [44] M. Grötschel and W. R. Pulleyblank. Weakly bipartite graphs and the max-cut problem. *Operations Research Letters*, 1(1):23–27, 1981.
- [45] F. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, 4(3):221–225, 1975.
- [46] F. Harary. On the notion of balance of a signed graph. *The Michigan Mathematical Journal*, 2(2):143–146, 1953.
- [47] D. S. Johnson. The NP-completeness column: an ongoing guide. *Journal of Algorithms*, 6(3):434–451, 1985.
- [48] R. M. Karp. *Reducibility among combinatorial problems*. Springer, 1972.

- [49] S. Klein, L. Faria, I. Sau, R. Sucupira, and U. Souza. On colored edge cuts in graphs. In Sociedade Brasileira de Computação, editor, *Primeiro Encontro de Teoria da Computação (ETC)*, Julho 2016.
- [50] M. Mitzenmacher and E. Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.
- [51] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*, volume 31. Oxford University Press, 2006.
- [52] G.I. Orlova and Ya. G. Dorfman. Finding the maximum cut in a planar graph. *Engineering Cybernetics*, 10(3):502–506, 1975.
- [53] S. Poljak and D. Turzík. A polynomial algorithm for constructing a large bipartite subgraph, with an application to a satisfiability problem. *Can. J. Math*, 34(3):519–524, 1982.
- [54] S. Poljak and D. Turzík. A polynomial time heuristic for certain subgraph optimization problems with guaranteed worst case bound. *Discrete Mathematics*, 58(1):99–104, 1986.
- [55] S. Poljak and Z. Tuza. Maximum cuts and large bipartite subgraphs. *DIMACS Series*, 20:181–244, 1995.
- [56] S. Porschen and E. Speckenmeyer. Algorithms for variable-weighted 2-sat and dual problems. In *Theory and Applications of Satisfiability Testing–SAT 2007*, pages 173–186. Springer, 2007.
- [57] R. Rizzi. NOTE - on minimizing symmetric set functions. *Combinatorica*, 20(3):445–450, 2000.
- [58] N. Robertson and P. D. Seymour. Graph minors. xiii. the disjoint paths problem. *Journal of combinatorial theory, Series B*, 63(1):65–110, 1995.
- [59] D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on computing*, 5(2):266–283, 1976.
- [60] V. Santos and U. Souza. Uma introdução à complexidade parametrizada. In Sociedade Brasileira de Computação (SBC), editor, *Anais da 34a. Jornada de Atualização em Informática JAI 2015*, pages 232–273. XXXV Congresso da Sociedade Brasileira de Computação, Julho 2015.
- [61] T. J. Schaefer. The complexity of satisfiability problems. In *Proc. 10th STOC*, pages 216–226, 1978.

- [62] R. Sucupira, L. Faria, and S. Klein. A complexidade do problema do corte máximo para grafos fortemente cordais. In Sociedade Brasileira de Pesquisa Operacional, editor, *Anais do XLV SBPO*, pages 2979–2988. XLV Simpósio Brasileiro de Pesquisa Operacional, Setembro 2013.
- [63] R. Sucupira, L. Faria, and S. Klein. Approximation algorithms for simple max cut of split graphs. In Sociedade Brasileira de Pesquisa Operacional, editor, *Anais do XLVI SBPO*, pages 3230–3235. XLVI Simpósio Brasileiro de Pesquisa Operacional, Setembro 2014.
- [64] J. L. Szwarcfiter. *Grafos e algoritmos computacionais*, volume 2. Campus, 1988.
- [65] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 13(3):566–579, 1984.
- [66] N. Van Ngoc and Z. Tuza. Linear-time approximation algorithms for the max cut problem. *Combinatorics, Probability and Computing*, 2(02):201–210, 1993.
- [67] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.
- [68] V. V. Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.
- [69] D. West. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.
- [70] M. Yannakakis. Node-and edge-deletion np-complete problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 253–264. ACM, 1978.
- [71] P. Zhang and B. Fu. The label cut problem with respect to path length and label frequency. *Theoretical Computer Science*, 648:72–83, 2016.

# Índice Remissivo

- $(s, t)$ -caminho, 71  
 $(s, t)$ -corte, 71  
 $(s, t)$ -grafo, 71  
 $C_{\text{ext}}$ , 53  
 $C_{\text{int}}$ , 53  
 $G - S$ , 48  
 $W[t]$ , 10  
 $W[t]$ -completo, 10  
 $W[t]$ -difícil, 10  
 $mc(G)$ , 11  
  
algoritmo, 6  
algoritmo de aproximação, 8  
algoritmo de OU-composição, 74  
algoritmo determinístico, 6  
algoritmo polinomial, 6  
algoritmo randomizado, 6, 12  
aresta marcada, 62  
arestas, 3  
arestas de corte, 10  
arestas múltiplas, 3  
arestas paralelas, 3  
  
Barahona, 15  
biclique, 4  
bipartido, 11  
bipartido completo, 4  
bloco-caminho, 53  
bloco-folha, 54  
Bodlaender e Jansen, 16, 29, 36  
  
caminho, 4  
ciclo, 4  
ciclo ímpar, 4  
  
ciclo hamiltoniano, 4  
ciclo negativo, 49  
ciclo par, 4  
ciclo positivo, 49  
classe  $P$ , 6  
classe de complexidade, 8  
classe hereditária, 5  
clique, 4, 35  
cliquetree, 24  
cobertura de conjuntos, 75  
cobertura de vértices, 76  
COLORFUL CUT, 74  
Coluna Johnson, 15  
complexidade clássica, 6  
complexidade parametrizada, 47  
componente conexa, 5  
comprimento do caminho, 4  
conjunto independente, 4, 35  
conjunto maximal, 5  
conjunto minimal, 5  
coNP, 6  
Cook, 14  
corda, 23  
cordal, 23  
corte de arestas, 10  
corte máximo, 11  
corte máximo simples, 11  
Crowston, 47  
  
derandomização, 14  
  
Edwards, 14  
Edwards-Erdős bound, 14, 47  
Erdős, 1, 12, 14

esquema de eliminação forte, 26  
 estrela, 4  
 exterior de um bloco, 53  
 fortemente cordal, 25  
 FPT, 9, 47, 50  
 FPT redução, 9  
 função de incidência, 3  
 função peso, 1  
 Garey e Johnson, 15  
 Garey, Johnson, Stockmeyer, 14, 18  
 Goemans e Williamson, 1, 15  
 grafo, 3  
 grafo  $(k, n)$ -split, 35  
 grafo  $(k, n)$ -split cheio, 39  
 grafo  $(k, n)$ -split cheio múltiplo, 40  
 grafo  $(k, n)$ -split cheio simples, 40  
 grafo  $(r, \ell)$ , 48  
 grafo  $(V_1, V_2)$ -balanceado, 45  
 grafo  $k$ -split, 35  
 grafo balanceado, 45  
 grafo bipartido, 4  
 grafo bloco, 50  
 grafo completo, 4  
 grafo conexo, 5  
 grafo desconexo, 5  
 grafo nulo, 4  
 grafo simples, 3  
 grafo sinalizado, 45  
 grafo split, 35  
 grafos  $k$ -regulares, 4  
 grafos cúbicos, 4  
 grafos fracamente bipartidos, 15  
 grau de um vértice, 4  
 Grötschel e Nemhauser, 15  
 Grötschel e Pulleyblank, 15  
 Hadlock, 1, 14  
 hiperaresta, 5  
 hipergrafo, 5  
 instância, 5  
 interior de um bloco, 53  
 Karp, 14, 16  
 laço, 3  
 método probabilístico, 12  
 MAX 2-SAT, 18  
 MAX COLORED CUT, 73  
 MAX CUT, 1, 15, 16  
 MIN COLORED  $(s, t)$ -CUT, 72  
 MIN COLORED CUT, 72  
 multigrafo, 3  
 multiplicidade, 40  
 núcleo, 10, 47, 50  
 núcleo polinomial, 50  
 NP, 6  
 NP-completo, 6, 14, 36  
 NP-difícil, 6  
 odd cycle transversal, 100  
 ordem de complexidade, 6  
 Orlova e Dorfman, 1, 14  
 parâmetro, 8, 47  
 PARTIÇÃO, 16  
 partição, 10  
 partição induzida, 10  
 peso do corte, 11  
 Poljak e Turzík, 48  
 Poljak e Tuza, 18  
 ponte, 5  
 PRAS, 8  
 problema computacional, 5  
 problema de decisão, 5  
 Problema do Carteiro Chinês, 15  
 problema parametrizado, 8  
 problemas de otimização, 7

pseudo-polynomial, 83  
PTAS, 8  
  
redução polinomial, 6  
regra segura, 52  
regra válida, 55  
regras de redução, 51  
  
S-redução, 7  
SAT, 14  
SET COVER, 75  
SIGNED MAX CUT, 46  
SIMPLE MAX CUT, 1, 11, 15, 18, 36  
sol, 26  
span, 84  
subgrafo, 4  
subgrafo induzido, 4  
  
vértice isolado, 4  
vértice marcado, 62  
vértice simples, 27  
vértice simplicial, 24  
vértice universal, 4  
vértice-de-caminho, 54  
vértices, 3  
VERTEX COVER, 76  
vizinhança aberta, 3  
vizinhança fechada, 3  
  
W hierarquia, 10  
  
Yannakakis, 21