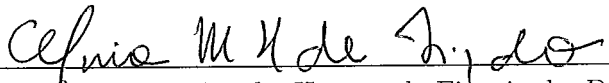


CORTES-ESTRELA E CORTES-CLIQUE SANDUÍCHE

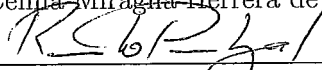
Rafael Bernardo Teixeira

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA
COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE
ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO
DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE
SISTEMAS E COMPUTAÇÃO.

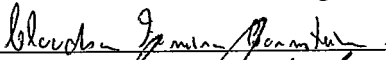
Aprovada por:



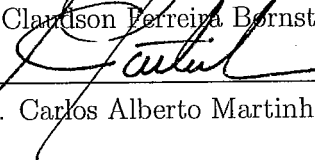
Prof.^a Celina Miraglia-Herrera de Figueiredo, D.Sc.



Prof. Renato Portugal, D.Sc.



Prof. Claudson Ferreira Bornstein, Ph.D.



Prof. Carlos Alberto Martinhon, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

JANEIRO DE 2004

TEIXEIRA, RAFAEL BERNARDO

Cortes-estrela e cortes-clique sanduíche
[Rio de Janeiro] 2004

VII, 35 p., 29,7 cm (COPPE/UFRJ, M.Sc.,
Engenharia de Sistemas e Computação, 2004)

Tese – Universidade Federal do Rio de
Janeiro, COPPE

1 - Grafos Sanduíche

2 - NP-Compleitude

I. COPPE/UFRJ II. Título (série)

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

CORTES-ESTRELA E CORTES-CLIQUE SANDUÍCHE

Rafael Bernardo Teixeira

Janeiro/2004

Orientadora: Celina Miraglia Herrera de Figueiredo

Programa: Engenharia de Sistemas e Computação

Este trabalho descreve um estudo sobre problemas sanduíche. Problemas sanduíche são generalizações de problemas de reconhecimento de uma propriedade Π . Enquanto em problemas de reconhecimento comum existe apenas um grafo como entrada, em problemas sanduíche existem dois grafos e procura-se por um terceiro grafo, intermediário entre os dois grafos de entrada, satisfazendo a propriedade Π . Apresentamos dois resultados originais sobre propriedades cujo reconhecimento é polinomial: corte-estrela e corte-clique. Mostramos que esses problemas despertam interesse para serem estudados em sua forma sanduíche. Propomos um algoritmo polinomial de complexidade $O(n^3)$ para o problema sanduíche para corte-estrela, fazendo algumas modificações num algoritmo conhecido simples de reconhecimento. Por fim, propomos uma prova de NP -completude, reduzindo o problema 1-EM-3 3SAT (sem literais negativos) para o problema sanduíche para corte-clique, notando a relação deste problema com o reconhecimento de corte estável, problema já conhecido NP -completo.

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

STAR CUTSET AND CLIQUE CUTSET SANDWICH

Rafael Bernardo Teixeira

January/2004

Advisor: Celina Miraglia Herrera de Figueiredo

Department : Computing and Systems Engineering

This work describes a study on graph sandwich problems. Sandwich problems generalize graph recognition problems with respect to a property Π . A recognition problem has a graph as input, whereas a sandwich problem has two graphs as input. In a sandwich problem, we look for a third graph, whose edge set lies between the edge sets of two given graphs. This third graph is required to satisfy a property Π . We present two original sandwich results corresponding to two known polynomial recognition problems: star cutset and clique cutset. We show these two graph decomposition problems are of interest with respect to sandwich problems. We propose a polynomial algorithm of $O(n^3)$ -time for star cutset sandwich problem. We propose an NP -completeness transformation from 1-IN-3 3SAT (with no negative literals) to clique cutset sandwich problem.

Índice

1	Introdução	1
2	Corte-Estrela Sanduíche	5
2.1	Definições	5
2.2	O Problema	6
2.3	O Algoritmo	8
2.4	Exemplos	12
3	Corte-Clique Sanduíche	17
3.1	Motivação	17
3.2	NP-Compleitude	18
3.3	Exemplo	32
4	Conclusões	36
4.1	Trabalhos futuros	37

Lista de Figuras

1.1	Apenas o grafo G possui <i>Corte-Clique</i>	3
1.2	Apenas o grafo G possui <i>Corte-Estrela</i>	3
2.1	<i>Corte-Estrela</i> com centro no vértice v	5
2.2	v 1-domina w	6
2.3	Completude: caso 1	10
2.4	Completude: caso 2	10
2.5	Completude: caso 3	11
2.6	Primeiro Exemplo	12
2.7	v não é centro de <i>corte-estrela</i>	13
2.8	v é universal e centro de <i>corte-estrela</i>	14
2.9	v e sua vizinhança desconectam o grafo	14
2.10	v é centro de <i>corte-estrela</i> pois v 1-domina w	14
2.11	Segundo Exemplo	15
2.12	v é universal, mas não centro de <i>corte-estrela</i>	15
2.13	v é <i>corte-estrela</i> que desconecta a de b	16
3.1	Subgrafo Cláusula C_j	19
3.2	Subgrafo Literal X_i e relações	20
3.3	Subgrafo Cláusula Desconexo	23
3.4	Subgrafo Literal Desconexo	23
3.5	Relações dos vértices k , s_1 e s_2	24

3.6	O vértice s_2 mantém a conectividade	25
3.7	O vértice s_1 mantém a conectividade	25
3.8	Os nove caminhos entre s_1 e s_2	26
3.9	O vértice $c_{j1} \in K'$	27
3.10	Os vértices b_{j2} e $b_{j3} \in K'$	28
3.11	Os vértices t_{j1} , t'_{j1} e $t_{j2} \in K'$	28
3.12	O vértice $b'_{j1} \in K'$	29
3.13	Os vértices c_{j2} e $c_{j3} \in K'$	29
3.14	Diagrama representando os componentes separando s_1 de s_2 .	29
3.15	Grafo G'	33
3.16	Exemplo 1 - Arestas Forçadas	33
3.17	Exemplo 1 - Arestas Proibidas	34
3.18	O grafo não possui <i>corte-clique</i>	35

Capítulo 1

Introdução

Os problemas sanduíche foram introduzidos em 1995 por Golumbic et al. [8]. Neste artigo, as propriedades consideradas se referem às subclasses de grafos perfeitos (*split*, cografo, intervalo, comparabilidade). Posteriormente foram consideradas propriedades relacionadas à decomposição, tais como conjunto homogêneo [2], partição *join* [6] entre outras. Recentemente, duas teses consideraram propriedades relacionadas à decomposição: em [4, 5] foram considerados os grafos (k, l) , uma generalização dos grafos *split*, e em [7] foram considerados algoritmos para decomposição usando conjunto homogêneo. Na presente dissertação consideramos também as decomposições.

Existem vários modos de definir um problema sanduíche, todos equivalentes. Daremos preferência ao seguinte: Dados dois grafos $G_1 = (V, E_1)$ e $G_2 = (V, E_2)$, dizemos que G_1 é subgrafo gerador de G_2 se $E_1 \subseteq E_2$. Dado mais um grafo $G = (V, E)$, dizemos ainda que G é um grafo sanduíche de G_1 e G_2 , se $E_1 \subseteq E \subseteq E_2$. Com isso podemos definir *Problemas de Grafo Sanduíche para a Propriedade II*:

PROBLEMA DE GRAFO SANDUÍCHE PARA PROPRIEDADE II

ENTRADA: Dois grafos $G_1(V, E_1)$ e $G_2(V, E_2)$, onde $E_1 \subseteq E_2$.

QUESTÃO: Existe um grafo sanduíche $G(V, E)$ para o par G_1, G_2 ,

que satisfaça a propriedade Π ?

Equivalentemente, podemos definir o mesmo problema da seguinte forma:

PROBLEMA DE GRAFO SANDUÍCHE PARA PROPRIEDADE Π

ENTRADA: Uma tripla de conjuntos (V, E_1, E_3) , onde $E_1 \cap E_3 = \emptyset$.

QUESTÃO: Existe um grafo $G(V, E)$, onde $E_1 \subseteq E$ e $E \cap E_3 = \emptyset$, que satisfaça a propriedade Π ?

As arestas em E_1 são chamadas forçadas ou obrigatórias, as arestas de E_3 são chamadas de proibidas, e as arestas de $E_2 \setminus E_1$ são chamadas de opcionais.

Problemas sanduíche generalizam problemas comuns de reconhecimento. Porém, não são todos os tipos de propriedade que despertam interesse. Se o problema de reconhecimento para uma propriedade Π já for NP -Completo, o problema sanduíche correspondente também o será. Há, também, duas características das propriedades que determinam se seu correspondente problema sanduíche é interessante: ancestralidade e hereditariedade.

Dizemos que uma propriedade é ancestral quando se um grafo G possui essa propriedade, todo supergrafo de G também possui. Por exemplo: conectividade. Dizemos que uma propriedade é hereditária quando se um grafo G possui essa propriedade, todo subgrafo de G também possui. Por exemplo: planaridade. Propriedades ancestrais ou hereditárias não são interessantes quando tratamos de problemas sanduíche. Nesse casos, quando uma propriedade for hereditária, bastaria testar G_1 , e quando propriedade for ancestral, bastaria testar G_2 .

Dado um grafo $G(V, E)$, um corte é subconjunto $S \subset V$ tal que a remoção dos vértices em S do grafo G produz o grafo desconexo $G \setminus S$. Caso S seja uma clique, isto é, S induza um grafo completo em G , dizemos que S é um *corte-clique*. Caso S contenha v adjacente em G a todos os outros vértices

de S , dizemos que S é um *corte-estrela*. Observe que um *corte-clique* é um *corte-estrela*.

As duas propriedades estudadas nesta dissertação correspondem a problemas de reconhecimento polinomiais. Whitesides [12] apresentou um algoritmo polinomial para o reconhecimento de *corte-clique* em um grafo G . Chvátal [3] provou um teorema que leva a um algoritmo polinomial para o reconhecimento de *corte-estrela* em um grafo G . Como podemos ver nas Figuras 1.1 e 1.2, as propriedades *Corte-Clique* e *Corte-Estrela* não são nem ancestrais, nem hereditários.

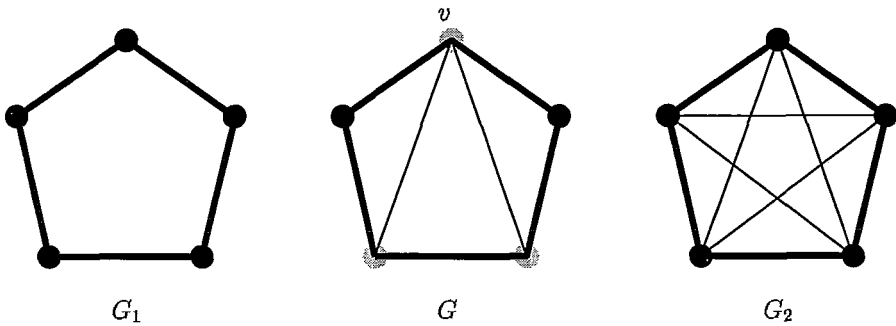


Figura 1.1: Apenas o grafo G possui *Corte-Clique*

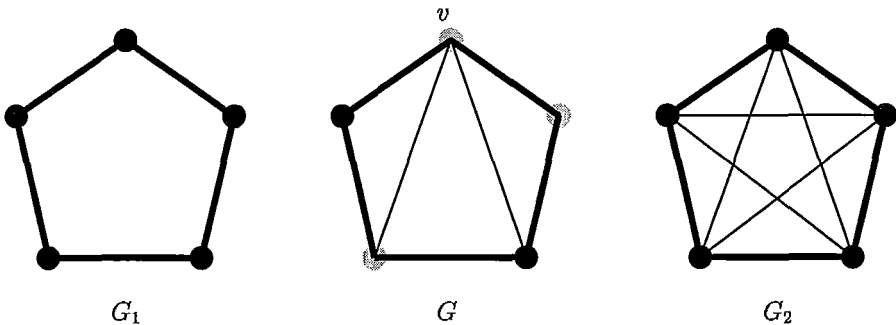


Figura 1.2: Apenas o grafo G possui *Corte-Estrela*

Nesta dissertação, apresentamos um algoritmo polinomial para o pro-

blema de grafo sanduíche para *corte-estrela* e uma prova de *NP*-completude para o problema de grafo sanduíche para *corte-clique*.

Capítulo 2

Corte-Estrela Sanduíche

2.1 Definições

Definimos um *corte-estrela* num grafo G como um conjunto de vértices C , tal que $G \setminus C$ é desconexo e algum vértice v de C é adjacente em G a todos os outros vértices de C , como mostrado na Figura 2.1. A esse vértice v chamamos de *centro* do *corte-estrela*.

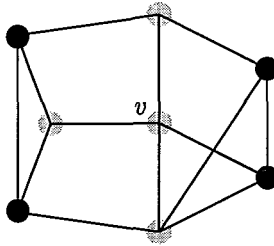


Figura 2.1: *Corte-Estrela* com centro no vértice v

Precisamos apresentar uma definição, bem adequada para problemas sanduíche, baseada no conceito de dominância. Sejam dois grafos conexos $G_1(V, E_1)$ e $G_2(V, E_2)$, onde $E_1 \subseteq E_2$, e dois vértices v, w . Denotamos por $N_1(v)$ o conjunto de vértices vizinhos a v no grafo G_1 . Analogamente, $N_2(v)$, representa o conjunto de vértices vizinhos a v em G_2 . Com isso, dizemos que v 1-domina w se $N_1(w) \subseteq N_2(v) \cup \{v\}$. Dizemos também que v é universal em

G_2 se $V = N_2(v) \cup \{v\}$. A Figura 2.2 mostra um exemplo de 1-dominância. Nesta figura, as arestas de E_1 aparecem mais escuras, enquanto que as arestas de $E_2 \setminus E_1$ aparecem menos escuras.

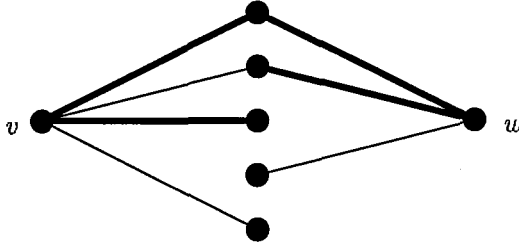


Figura 2.2: v 1-domina w

2.2 O Problema

Chvátal [3] apresentou um algoritmo polinomial para reconhecimento de *corte-estrela* em um grafo. Vamos apresentar uma generalização desse algoritmo para o problema sanduíche correspondente:

CORTE-ESTRELA SANDUÍCHE

ENTRADA: Dois grafos $G_1(V, E_1)$ e $G_2(V, E_2)$, onde $E_1 \subseteq E_2$.

QUESTÃO: Existe um grafo sanduíche $G(V, E)$ para o par G_1, G_2 , que contenha um *corte-estrela*?

O teorema a seguir leva a um algoritmo polinomial para resolver o problema em questão.

Teorema 2.1 *Dados dois grafos $G_1 = (V, E_1)$ e $G_2 = (V, E_2)$ onde $E_1 \subseteq E_2$, existe um grafo sanduíche $G = (V, E)$ para o par G_1, G_2 se e somente se pelo menos uma das seguintes situações ocorrer no par G_1, G_2 :*

- (i) Existe um vértice v tal que o conjunto de vértices formado por $V \setminus (N_2(v) \cup v)$ induza um subgrafo desconexo em G_1 .
- (ii) Existem pelo menos dois vértices a, b não-adjacentes em G_1 e existem dois vértices v, w , adjacentes em G_2 , tais que v 1-domina w .

Prova. Primeiramente provaremos a parte “se”. Para isso precisamos mostrar que as propriedades (i) e (ii) nos levam a um grafo sanduíche $G(V, E)$, que possui um *corte-estrela*. Supondo que a propriedade (i) ocorra, podemos construir G da seguinte forma: o vértice v e sua vizinhança $N_2(v)$ formam o conjunto C . Adicionando a G_1 as arestas de $E_2 \setminus E_1$ que são incidentes a v , temos o grafo G , onde C é um *corte-estrela*. Agora, supondo que a propriedade (ii) ocorra, temos duas possibilidades. A primeira surge quando existe pelo menos um vértice u , tal que $u \notin N_2(v) \cup \{v\}$. Podemos formar o grafo G da seguinte forma: o vértice v e sua vizinhança $N_2(v)$, com exceção de w , formam o conjunto C . Adicionando a G_1 as arestas de $E_2 \setminus E_1$ que são incidentes a v , mas não a w , temos o grafo G , com C como um *corte-estrela*. De fato, como v 1-domina w , temos que qualquer caminho em G que ligue u a w contém algum vértice de C . A segunda possibilidade acontece quando v é vizinho de todos os outros vértices em G_2 . Se $v \neq a, b$, o *corte-estrela* C contém $\{v\} \cup [N_2(v) \setminus \{a, b\}]$. O grafo sanduíche G é obtido de G_1 adicionando as arestas de $E_2 \setminus E_1$ incidentes a v e não adjacentes aos vértices a ou b . Finalmente, caso $v = a$, se $V \setminus \{a\}$ não for uma clique de G_1 , então o *corte-estrela* $C = \{a\} \cup [N_2(a) \setminus \{x, y\}]$, onde x não é adjacente a y em G_1 , se $V \setminus \{a\}$ for uma clique de G_1 , então o *corte-estrela* $C = \{x\} \cup [N_2(x) \setminus \{a, b\}]$, onde $x \neq a, b$.

Agora, provaremos a parte “somente se”. Para essa parte, precisamos mostrar que a existência de um grafo sanduíche G nos leva a pelo menos uma das propriedades (i) ou (ii). Seja, então, um grafo sanduíche G , onde C

é um *corte-estrela* de G . Seja, também, v o vértice central do *corte-estrela* C . Como C é um corte, separa G em componentes V_1, \dots, V_n , onde $n \geq 2$. Suponha que existem $i \neq j$ tais que: $v_i \in V_i \setminus N_2(v)$ e $v_j \in V_j \setminus N_2(v)$. Qualquer caminho em G_1 de v_i a v_j contém $x \in C \subseteq \{v\} \cup N_2(v)$. Como $v_i, v_j \notin N_2(v)$, temos que $G_1 \setminus \{v\} \cup N_2(v)$ é desconexo, o que é a propriedade (i). Caso contrário, v é adjacente em G_2 a todos os vértices de pelo menos um componente V_i , o que implica no fato de que v 1-domina todos os vértices de V_i . Temos, então, a propriedade (ii). ■

Assim, mostramos que o teorema é verdadeiro. A seguir, apresentamos o algoritmo obtido a partir do teorema visto e sua análise de complexidade, mostrando que o algoritmo é polinomial.

2.3 O Algoritmo

Eis o algoritmo:

- 1: para todo $v \in V$ faça
- 2: $C \leftarrow v \cup N_2(v)$
- 3: se $(G_1 \setminus C)$ for desconexo então
- 4: $O \leftarrow E_2 \setminus E_1$ incidente a v
- 5: $G \leftarrow G(V, E_1 \cup O)$
- 6: retorna G
- 7: senão se $C = V$ então
- 8: para todo $a, b \in N_2(v)$ tais que $a \neq b$ faça
- 9: se $(a, b) \notin E_1$ então
- 10: $C \leftarrow C \setminus \{a, b\}$
- 11: $O \leftarrow E_2 \setminus E_1$ incidente a v
- 12: $G \leftarrow G(V, E_1 \cup O)$

```

13:         retorna  $G$ 
14:   senão
15:     para todo  $w \in N_2(v)$  faça
16:       se  $v$  1-domina  $w$  então
17:          $C \leftarrow C \setminus \{w\}$ 
18:          $O \leftarrow E_2 \setminus E_1$  incidente a  $v$ 
19:          $G \leftarrow G(V, E_1 \cup O)$ 
20:       retorna  $G$ 
21: retorna “Não existe grafo sanduíche  $G$ ”

```

Teorema 2.2 *O algoritmo verifica corretamente e em tempo $O(n^3)$ se um par (G_1, G_2) admite grafo sanduíche G com corte-estrela.*

Prova. É claro que nas linhas 6, 13, 20 o grafo G retornado pelo algoritmo é grafo sanduíche para (G_1, G_2) e tem C como corte-estrela.

Por outro lado, seja G , seja G um grafo sanduíche para (G_1, G_2) com corte-estrela C . Seja v o centro do corte. Logo existem pelo menos duas componentes V_i, V_j de $G \setminus C$ tais que $(v_i, v_j) \notin E_1$, para todo $v_i \in V_i, v_j \in V_j$.

Caso 1: Existem $a \in V_i, b \in V_j, a, b \notin N_2(v)$, como mostrado na Figura 2.3. O algoritmo, ao testar o vértice v , retorna um grafo sanduíche na linha 6, pois $G_1 \setminus N_2(v) \cup \{v\}$ é desconexo.

Caso 2: $V_i \subseteq N_2(V)$ e existe $a \in V_j \setminus N_2(v)$, como mostrado na Figura 2.4. O algoritmo, ao testar o vértice v retorna um grafo sanduíche na linha 20, pois v 1-domina qualquer $v_i \in V_i$.

Caso 3: $V = N_2(v) \cup v$, como mostrado na Figura 2.5. O algoritmo retorna um grafo sanduíche na linha 13, pois v é universal e existe um par de vértices $(a, b), a \in V_i, b \in V_j$, tal que a aresta (a, b) não pertence a E_1 .

Analisando o algoritmo acima, podemos ver que ele é polinomial. Seja $n = |V|$, o número de vértices em G_1 e em G_2 . A linha 1 indica um teste para

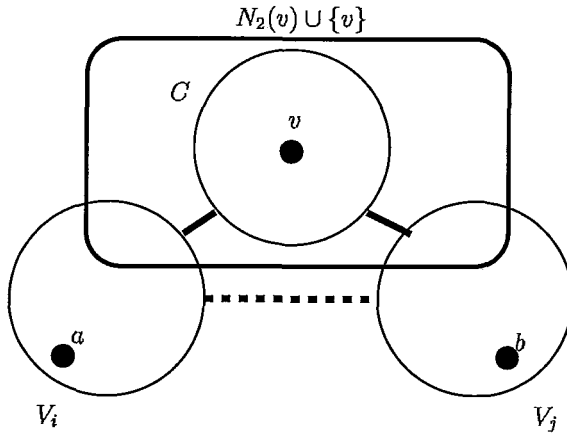


Figura 2.3: Completude: caso 1

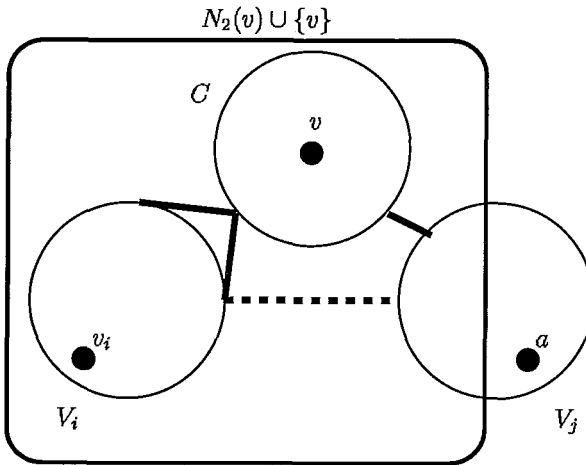


Figura 2.4: Completude: caso 2

cada vértice $v \in V$, garantindo complexidade pelo menos $O(n)$. Na linha 2, procura-se pelos vizinhos de v . Essa operação também é $O(n)$. Da linha 3 à linha 19, são testadas até três condições para v . A primeira, na linha 3, testa a conectividade do grafo $G_1 \setminus C$, o que pode ser feito por uma busca em $O(n^2)$. A segunda, na linha 7, compara os conjuntos C e V , bastando verificar a cardinalidade: $O(1)$. Seguindo essa operação, na linha 8, procura-se por

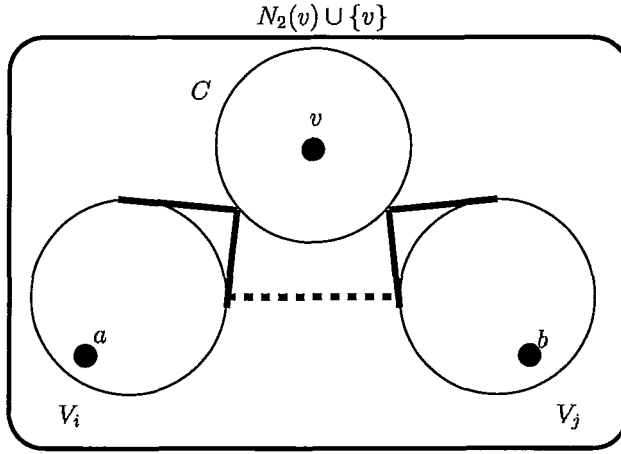


Figura 2.5: Completude: caso 3

cada par de vértices de $N_2(v)$ não adjacentes em G_1 . Como há $O(n)$ vértices em $N_2(v)$, são $O(n^2)$ pares de vértices. O teste de pertinência, realizado na linha 9, tem complexidade constante, desde que sejam utilizadas estruturas de dados adequadas. A terceira condição (linha 14) ocorre quando as outras duas falham. Desta vez, são testados os vértices vizinhos em G_2 a v , $O(n)$ vértices. Na linha 16, o teste de 1-dominância tem complexidade $O(n)$. A operação de geração do grafo G , caso exista, tem complexidade $O(n)$.

Para a complexidade total do algoritmo, observamos que cada condição testada pelo algoritmo tem complexidade diferente. A primeira condição, da linha 3 a 6, tem complexidade $O(n^2)$, a segunda condição da linha 7 a 13, $O(n^2)$, na verdade, $O(1) + O(n^2).O(1)$ e a terceira, da linha 14 a 20, como a anterior, $O(n^2)$, mas, por motivos diferentes, originado de $O(n).O(n)$. Toda vez que a primeira condição for satisfeita, é encontrado um grafo sanduíche G . Esta situação não ocorre quando as outras condições, nas linhas 7 e 14, são satisfeitas. Em vista disso, a complexidade de pior caso para o algoritmo ocorre quando a segunda ou a terceira condições são sempre satisfeitas, mas

falhando em encontrar G . A complexidade para cada vértice então é $O(n^2)$, não importando qual das duas condições foi satisfeita, pois a complexidade é a mesma. Isso mostra que a complexidade total do algoritmo é $O(n^3)$. ■

2.4 Exemplos

Mostraremos agora exemplos de execução do algoritmo. Nesses exemplos, analisaremos casos onde o algoritmo procura por um *corte-estrela* com centro em um vértice $v \in V$. Apresentamos dois exemplos (Figuras 2.6 e 2.11), onde estão representados os vértices rotulados a serem analisados, as arestas forçadas (pertencentes a E_1), mais escuras nas figuras, e as arestas opcionais (pertencentes a $E_2 \setminus E_1$), mais claras nas figuras. Para podermos visualizar cada caso do algoritmo, propomos várias execuções, nas quais a ordem de visita aos vértices será diferente, visto que o algoritmo termina ao encontrar um *corte-estrela*.

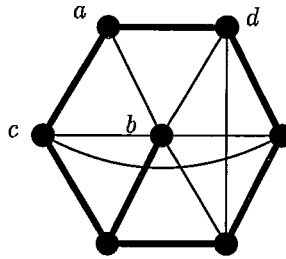


Figura 2.6: Primeiro Exemplo

Seja uma entrada para o algoritmo proposto de acordo com a Figura 2.6. Suponha que o vértice $v = a$ seja o primeiro a ser visitado pelo algoritmo. O primeiro teste do algoritmo (na linha 3) verifica se $G_1 \setminus v \cup N_2(v)$ é desconexo, podemos ver, na Figura 2.7, onde $v \cup N_2(v)$ aparecem como vértices brancos, que esse teste falha. Passamos então ao teste (na linha 7), que verifica se

v é universal em G_2 . Outra vez o teste falha, v não é universal. Por fim, chegamos ao último teste (na linha 14), que procura por um vértice w , tal que v 1-domina w . Esse último teste também falha, e temos que v não é centro de *corte-estrela*. O próximo vértice é $v = b$. Podemos ver, na Figura 2.8, que o primeiro teste falha, mas não o segundo: v é universal em G_2 . O algoritmo, agora, procura por dois vértices a, b , não adjacentes em G_1 . Existindo tais vértices, v é centro de *corte-estrela* desconectando a de b , como podemos ver na Figura 2.8, onde os vértices do corte estão marcados de branco.

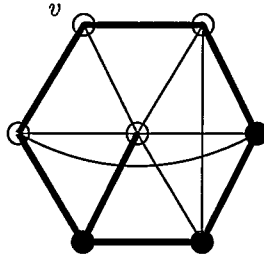


Figura 2.7: v não é centro de *corte-estrela*

A execução anterior terminou, pois foi encontrado um *corte-estrela*. Vamos começar uma outra execução do algoritmo, ainda usando o exemplo da Figura 2.6. Suponha agora que $v = c$ seja o primeiro vértice visitado pelo algoritmo. Podemos ver, na Figura 2.9, que v satisfaz a condição do primeiro teste: $G_1 \setminus v \cup N_2(v)$ é desconexo. Os vértices do corte estão em branco na Figura 2.9.

A última execução do algoritmo com o exemplo da Figura 2.6 tem como primeiro vértice $v = d$. Podemos ver, na Figura 2.10, que o primeiro teste falha ($G_1 \setminus v \cup N_2(v)$ é conexo), assim como o segundo (v não é universal em G_2). Porém, o terceiro teste encontra um vértice $w \in N_2(v)$ tal que v 1-domina w . Isso indica um *corte-estrela* que desconecta w dos outros vértices fora do corte.

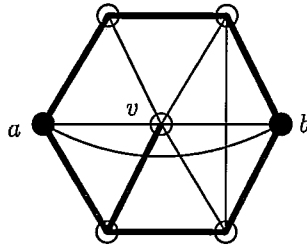


Figura 2.8: v é universal e centro de *corte-estrela*

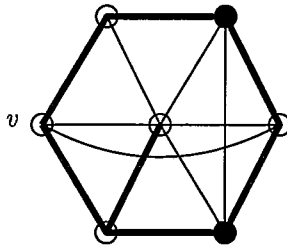


Figura 2.9: v e sua vizinhança desconectam o grafo

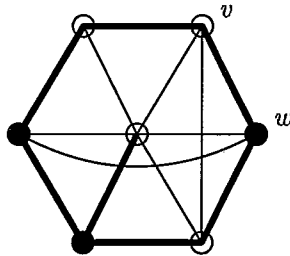


Figura 2.10: v é centro de *corte-estrela* pois v 1-domina w

Já mostramos todos os casos simples até agora: v e sua vizinhança formam *corte-estrela*, v é universal em G_2 , v 1-domina w , e um caso onde v não é centro de *corte-estrela*. Mostraremos agora um caso onde v é universal em G_2 mas não é centro de *corte-estrela*. Para tal, fornecemos ao algoritmo uma entrada de acordo com a Figura 2.11. Iniciaremos o algoritmo com o vértice $v = a$. Como v é universal em G_2 , o primeiro teste falha. O segundo

teste também falha, pois, ao procurar pelos vértices a, b não adjacentes em G_1 (linha 9), esses vértices não são encontrados, assim temos que v não é centro de *corte-estrela* (Figura 2.12). É interessante notar que, neste caso, há *corte-estrela*, como mostramos em seguida no segundo vértice visitado pelo algoritmo.

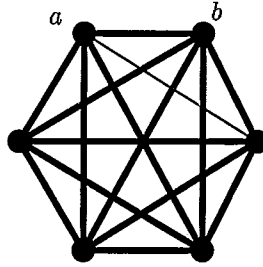


Figura 2.11: Segundo Exemplo

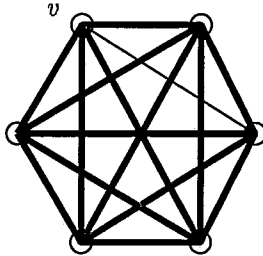


Figura 2.12: v é universal, mas não centro de *corte-estrela*

Escolhemos, agora, o vértice $v = b$ da Figura 2.11. Esse vértice v , assim como qualquer outro, é universal em G_2 . Temos então que v não passa no primeiro teste, mas, como podemos ver na Figura 2.13, v passa no segundo, pois existe um par de vértices a, b , tal que a e b são não adjacentes em G_1 . Observamos que tal par é único para essa entrada, e a razão pela qual o primeiro vértice na Figura 2.12 testado não é centro de *corte-estrela* nessa instância do algoritmo é justamente que o vértice fazia parte desse par.

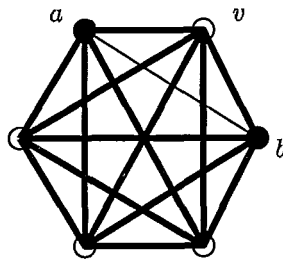


Figura 2.13: v é *corte-estrela* que desconecta a de b

Assim terminamos os exemplos para o algoritmo para o problema CORTE-ESTRELA SANDUÍCHE.

Capítulo 3

Corte-Clique Sanduíche

3.1 Motivação

O problema de reconhecimento de *corte-clique* em um grafo foi provado ser polinomial, com complexidade $O(n^3)$ em 1981 por Whitesides [12]. Em 2003, Kratsch et al. [10] reduziram a complexidade do problema para $O(n^{2.69})$. Parecia simples generalizar o algoritmo, como feito para o problema do *corte-estrela*, gerando um outro algoritmo mostrando a polinomialidade de *corte-clique* sanduíche.

Porém, foi notado que, considerando-se uma instância do problema sanduíche onde G_2 é o grafo completo, se G_1 contiver um corte estável C (um conjunto de vértices onde os vértices são dois a dois não adjacentes que também é um corte), necessariamente existirá um grafo G com um *corte-clique* C' , adicionando a C as arestas faltantes para se tornar uma clique. Klein [9] e Brandstädt [1] mostraram que o reconhecimento de corte estável em grafos é *NP*-Completo. Isso nos levou a sugerir que CORTE-CLIQUE SANDUÍCHE também fosse. Porém, essa singularidade não é suficiente para provar, o que nos levou à essa prova apresentada a seguir.

Para o problema de reconhecimento de corte estável, Brandstädt fez uma redução do problema de 1-EM-3 3SAT (sem literais negativos), provado ser

NP-Completo por Schaefer [11]:

1-EM-3 3SAT (sem literais negativos)

ENTRADA: Um conjunto X de n variáveis x_i e um conjunto C de m cláusulas de três variáveis $C_j = (c_{j1} \vee c_{j2} \vee c_{j3})$, onde $1 \leq i \leq n$ e $1 \leq j \leq m$.

QUESTÃO: Existe uma atribuição satisfazendo C tal que cada cláusula em C contenha exatamente uma variável verdadeira?

Apresentamos aqui uma redução desse mesmo problema de satisfabilidade para o problema do *corte-clique* sanduíche. Eis a definição do PROBLEMA DO CORTE-CLIQUE SANDUÍCHE:

PROBLEMA DO CORTE-CLIQUE SANDUÍCHE

ENTRADA: Uma tripla de conjuntos (V, E_1, E_3) , onde $E_1 \cap E_3 = \emptyset$.

QUESTÃO: Existe um grafo $G(V, E)$, onde $E_1 \subseteq E$ e $E \cap E_3 = \emptyset$, que contenha um *corte-clique*?

3.2 NP-Completo

Para fazer a redução de 1-EM-3 3SAT (sem literais negativos) para CORTE-CLIQUE SANDUÍCHE (CCS), vamos construir uma entrada G' de CCS baseado numa instância de 1-EM-3 3SAT (sem literais negativos). Observe que G' representa uma entrada para o problema sanduíche, sendo nele representadas as arestas forçadas (E_1) e as arestas proibidas (E_3). As arestas não representadas formam o conjunto $E_2 \setminus E_1$ de arestas opcionais. Embora G' contenha três tipos de arestas: forçadas, proibidas e opcionais, vamos nos referir a G' como um grafo por simplicidade. Representamos em G' as arestas forçadas (E_1) por linhas contínuas, as arestas proibidas (E_3) por linhas pontilhadas, e as arestas omitidas correspondem às arestas opcionais ($E_2 \setminus E_1$).

Para cada cláusula C_j , adicionaremos a G' um subgrafo cláusula C_j da seguinte forma de acordo com a Figura 3.1.

- Os vértices $c_{j1}, c_{j2}, c_{j3}, b_{j1}, b_{j2}, b_{j3}, b'_{j1}, b'_{j2}, b'_{j3}, t_{j1}, t_{j2}, t_{j3}, t'_{j1}, t'_{j2}, t'_{j3}$;
- As arestas forçadas $(c_{j1}, b_{j1}), (c_{j2}, b_{j2}), (c_{j2}, b'_{j2}), (c_{j1}, t_{j3}), (c_{j1}, t'_{j3}), (c_{j2}, t_{j1}), (c_{j2}, t'_{j1}), (c_{j3}, t_{j2}), (c_{j3}, t'_{j2}), (t_{j1}, t'_{j1}), (t_{j2}, t'_{j2}), (t_{j2}, t'_{j2}), (t_{j1}, b'_{j2}), (t_{j2}, b'_{j3}), (t_{j3}, b'_{j1}), (t'_{j1}, b'_{j3}), (t'_{j2}, b'_{j1}), (t'_{j3}, b'_{j2})$;
- As arestas proibidas $(c_{j1}, c_{j2}), (c_{j1}, c_{j3}), (c_{j2}, c_{j3}), (b_{j1}, b'_{j1}), (b_{j2}, b'_{j2}), (b_{j2}, b'_{j2}), (c_{j1}, b'_{j2}), (c_{j2}, b'_{j3}), (c_{j3}, b'_{j1}), (t'_{j1}, t'_{j2}), (t'_{j1}, t'_{j3}), (t'_{j2}, t'_{j3}), (t_{j1}, t'_{j2}), (t_{j2}, t'_{j3}), (t_{j3}, t'_{j1}), (t'_{j1}, b'_{j2}), (t'_{j2}, b'_{j3}), (t'_{j3}, b'_{j1})$.

Nesse subgrafo, os vértices c_{ja} correspondem às variáveis c_{ja} das cláusulas, onde $1 \leq j \leq m$ e $1 \leq a \leq 3$. Por conveniência, na Figura 3.1, omitimos o índice j .

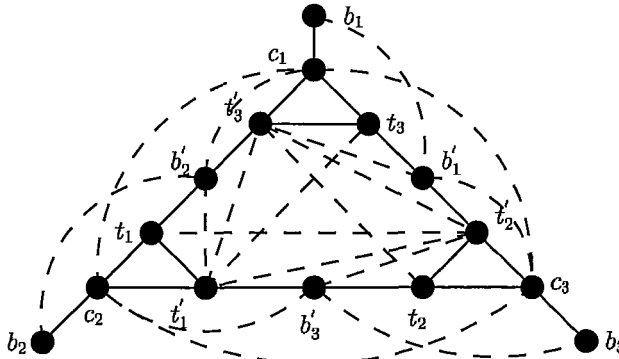


Figura 3.1: Subgrafo Cláusula C_j

Adicionaremos também os vértices auxiliares s_1, s_2 e k a G' . E, para cada literal x_i , um subgrafo literal X_i contendo os vértices x_i, x'_i e l_i . Todos esses vértices estão relacionados da seguinte forma:

- Arestas forçadas: $(k, s_1), (k, s_2), (x_i, s_1), (x'_i, s_2), (x_i, l_i), (x'_i, l_i)$;

- Arestas proibidas: $(s_1, s_2), (x_i, x'_i), (x_i, s_2), (x'_i, s_1), (k, l_i), (s_1, l_i), (s_2, l_i)$.

A Figura 3.2 ilustra a relação entre os vértices auxiliares e cada subgrafo literal. Os vértices x_i , onde $1 \leq i \leq n$, representam as variáveis x_i da entrada do problema de satisfabilidade referido.

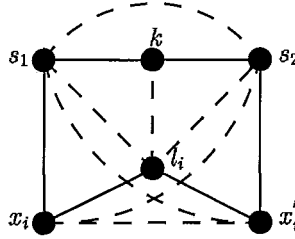


Figura 3.2: Subgrafo Literal X_i e relações

Com isso, concluímos a definição de V , conjunto de vértices de G' . Mas ainda precisamos de algumas arestas extras para conectar os subgrafos cláusula e o conjunto formado pelos vértices auxiliares e os subgrafos literal. Para isso adicionaremos a G' as seguintes arestas forçadas:

- (c_{ja}, s_1)
- (b_{ja}, s_2)
- (b'_{ja}, s_2)

onde $1 \leq a \leq 3$ e $1 \leq j \leq m$.

Na construção do grafo G' , não foi levada em consideração nenhuma relação entre o conjunto de cláusulas C e o conjunto de variáveis X . Essas relações estão representadas pelas arestas proibidas descritas a seguir:

- se $c_{ja} = x_i$, então (c_{ja}, x_i) e (b_{ja}, x'_i) são arestas proibidas;

- caso contrário, (c_{ja}, x_i) e (b_{ja}, x'_i) são arestas opcionais;

onde $1 \leq i \leq n$, $1 \leq a \leq 3$ e $1 \leq j \leq m$.

Para completar o conjunto de arestas proibidas temos:

- (c_{ja}, s_2) , (b_{ja}, s_1) e (b'_{ja}, s_1)
- (x_i, s_2) e (x'_i, s_1) , já mostradas na Figura 3.2
- (t_{ja}, s_1) , (t'_{ja}, s_1) , (t_{ja}, s_2) e (t'_{ja}, s_2)

onde $1 \leq i \leq n$, $1 \leq a \leq 3$ e $1 \leq j \leq m$.

Agora o grafo G' está pronto de acordo com uma instância do problema 1-EM-3 3SAT (sem literais negativos). Seja então uma atribuição verdade para as variáveis de X . A partir do grafo G' e dessa atribuição, vamos construir um subconjunto K de V de acordo com as seguintes instruções :

- se a variável x_i for verdadeira, $x'_i \in K$, caso contrário, $x'_i \notin K$;
- se a variável x_i for falsa, $x_i \in K$, caso contrário, $x_i \notin K$;
- se $x'_i \in K$ e $x_i = c_{ja}$, então $c_{ja} \in K$, caso contrário, $c_{ja} \notin K$;
- se $c_{j1} \in K$, $t_{j1}, t'_{j1}, t_{j2}, b'_{j1}, b_{j2}, b_{j3} \in K$;
- se $c_{j2} \in K$, $t_{j2}, t'_{j2}, t_{j3}, b'_{j2}, b_{j3}, b_{j1} \in K$;
- se $c_{j3} \in K$, $t_{j3}, t'_{j3}, t_{j1}, b'_{j3}, b_{j1}, b_{j2} \in K$;
- $k \in K$

onde $1 \leq i \leq n$, $1 \leq a \leq 3$ e $1 \leq j \leq m$.

Faremos agora a seguinte proposição:

Proposição 3.1 *K é uma clique de G_2 .*

Prova. Uma clique em G_2 possui apenas vértices ligados por arestas forçadas ou opcionais. Então, precisamos provar que não existem arestas proibidas entre os elementos de K . Como podemos ver nas Figuras 3.3 e 3.4, não existem dois vértices de K ligados por arestas proibidas nos subgrafos. Entre os subgrafos, também não existem vértices ligados por arestas proibidas, devido aos seguintes fatos:

- Os vértices $s_1, s_2, l_i \notin K$, logo, as arestas proibidas incidentes a s_1 , a s_2 ou a l_i não são arestas proibidas entre os elementos de K ;
- Por construção do conjunto K , temos que, se o vértice $x'_i \in K$ e a variável $x_i = c_{ja}$, então $c_{ja} \in K$. Porém, por construção do grafo G' , não existem arestas proibidas entre tais vértices;
- Por construção do conjunto K , temos que, se o vértice $x_i \in K$ e a variável $x_i = c_{ja}$, então $c_{ja} \notin K$. Como, neste caso $x'_i \notin K$, não temos a aresta proibida (x'_i, b_{ja}) entre vértices de K .

onde $1 \leq a \leq 3$ e $1 \leq j \leq m$.

Logo, o conjunto K é uma clique de G_2 . ■

Sabemos agora que K é uma clique de G_2 . Então vamos assumir que exista um outro subconjunto de vértices K' de G_2 , tal que K' não possua

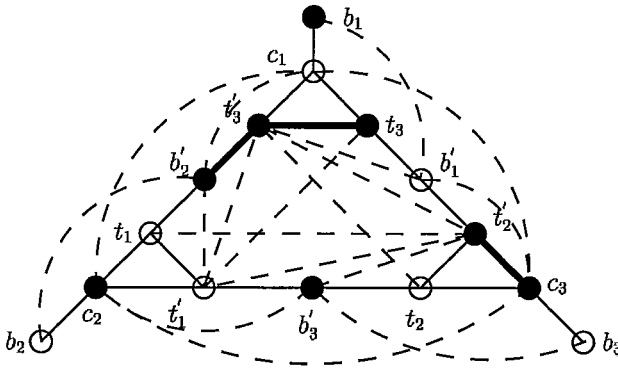


Figura 3.3: Subgrafo Cláusula Desconexo

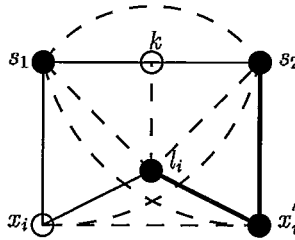


Figura 3.4: Subgrafo Literal Desconexo

aresta proibida e desconecte G_1 , independente de qualquer atribuição verdade. Vamos fazer algumas considerações sobre K' .

Proposição 3.2 *Em $G_1 \setminus K'$ os seguintes pares de vértices não pertencem a componentes conexas distintas: k e z , onde z pertence a um subgrafo cláusula ou z pertence a um subgrafo literal.*

Prova. Todo o caminho em G_1 entre um vértice de C_j e k , passa por s_1 ou s_2 . Em C_j , os vértices vizinhos a s_1 e s_2 são c_{ja} , b_{ja} e b'_{ja} . A Figura 3.5 mostra a relação entre esses vértices e k . Se $s_1 \in K'$, $s_2, b_{ja}, b'_{ja} \notin K'$ e, analogamente para $s_2 \in K'$. Dessa forma haverá sempre um caminho forçado de c_{ja} , b_{ja} ou b'_{ja} até k , passando ou por s_1 , ou por s_2 . Logo k e c_{ja} não pertencem a componentes distintas de $G_1 \setminus K'$ (analogamente k e b_{ja} , k e b'_{ja}).

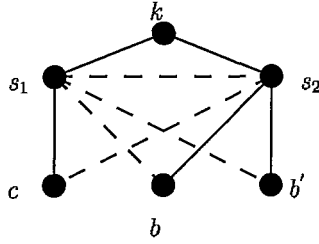


Figura 3.5: Relações dos vértices k , s_1 e s_2

Existem dois caminhos forçados entre x_i e k : x_i, s_1, k e x_i, l_i, x'_i, s_2, k . Podemos facilmente ver, pela Figura 3.2, que incluindo qualquer dos vértices internos de um caminho em K' , impede que o outro caminho forçado seja desfeito. Logo, k e x_i não pertencem a componentes distintas de $G_1 \setminus K'$. Chegamos à mesma conclusão para o par x'_i e k .

Para as provas de k e t_{ja} , k e t'_{ja} , k e l_i , usamos um argumento analogo.

■

Proposição 3.3 *Os vértices $s_1, s_2 \notin K'$.*

Prova. Suponhamos que $s_1 \in K'$. Isso impede que os vértices $s_2, x'_i, l_i, b_{ja}, b'_{ja}, t_{ja}$ e t'_{ja} pertençam a K' , permitindo apenas a presença em K' dos vértices k, x_i e c_{ja} . Mesmo se colocássemos todos esses vértices permitidos em K' , o vértice s_2 garantiria a conectividade de $G_1 \setminus K'$, como podemos ver na Figura 3.6.

Analogamente, se fosse $s_2 \in K'$, impediria que os vértices $s_2, x_i, l_i, c_{ja}, t_{ja}$ e t'_{ja} pertençam a K' , permitindo apenas a presença dos vértices k, x'_i, b'_{ja} e b_{ja} . Mesmo se colocássemos todos esses vértices permitidos em K' , o vértice s_1 garantiria a conectividade de $G_1 \setminus K'$, como podemos ver na Figura 3.7.

Unindo essas duas refutações, temos que a Proposição 3.3 é verdadeira.

■

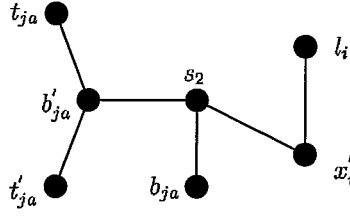


Figura 3.6: O vértice s_2 mantém a conectividade

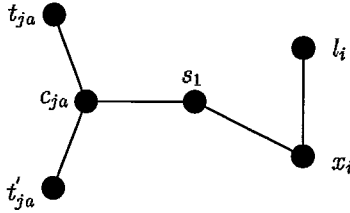


Figura 3.7: O vértice s_1 mantém a conectividade

Proposição 3.4 *O vértice $k \in K'$.*

Prova. Suponhamos que $k \notin K'$. Pela Proposição 3.3, os vértices s_1 e $s_2 \notin K'$. Logo, os vértices, k , s_1 e s_2 estão numa mesma componente de $G_1 \setminus K'$. Seja z um vértice de um outro componente de $G_1 \setminus K'$. Como z pertence a um subgrafo cláusula ou a um subgrafo literal, obtemos uma contradição à Proposição 3.2. ■

Proposição 3.5 *Em $G_1 \setminus K'$, s_1 e s_2 pertencem a componentes distintas.*

Prova. Se s_1 e s_2 estivessem no mesmo componente conexo de $G_1 \setminus K'$, $G_1 \setminus (K' \setminus \{k\})$ também seria desconexo, contrariando a Proposição 3.4. De fato, em $G_1 \setminus (K' \setminus \{k\})$, os vértices s_1 , s_2 e k estariam na mesma componente. Seja z um vértice de uma outra componente de $G_1 \setminus K'$. Qualquer caminho em G_1 entre k e z contém s_1 ou s_2 . Assim z também está numa outra componente de $G_1 \setminus (K' \setminus \{k\})$. ■

Sabemos, pelas Proposições 3.3, 3.4 e 3.5, que $k \in K'$ e que K' desconecta s_1 de s_2 . Podemos definir agora quais são os outros vértices pertencentes a K' . Temos em G_1 , n subgrafos literais X_i , $1 \leq i \leq n$, e m subgrafos cláusulas C_j , $1 \leq j \leq n$. Considerando os caminhos em G_1 entre s_1 e s_2 , temos três tipos de caminhos: caminhos com vértices internos pertencentes a um subgrafo cláusula, caminhos com vértices internos pertencentes a um subgrafo literal e o caminho s_1, k, s_2 cujo vértice interno k não pertence a subgrafo cláusula ou subgrafo literal.

Cada subgrafo literal X_i é formado pelos vértices x_i , x'_i e l_i . O vértice l_i desconectaria o caminho entre s_1 e s_2 , mas não pode estar em K' porque existe aresta proibida entre l_i e k , e o vértice k pertence a K' . Devemos, então, incluir em K' os vértices x_i ou x'_i , mas não ambos, por causa da aresta proibida entre eles.

Cada subgrafo cláusula C_j possui nove caminhos minimais (i. e. sem cordas) entre s_1 e s_2 . Esses caminhos, como podemos identificar na Figura 3.8, sempre estão entre um vértice c , ligado a s_1 , e um vértice b ou b' , ligados a s_2 . Precisamos mostrar como desconectar todos esses caminhos. Se

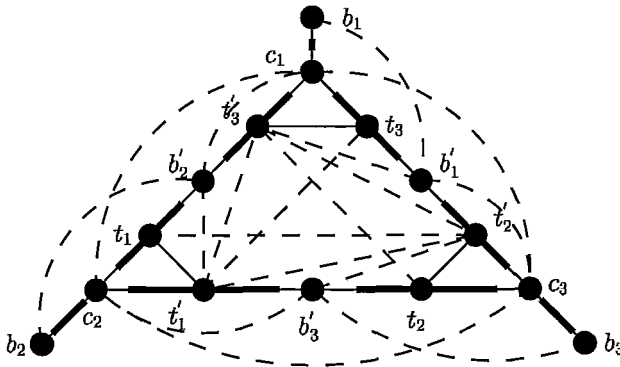


Figura 3.8: Os nove caminhos entre s_1 e s_2

incluirmos em K' todos os vértices c_{ja} , $1 \leq a \leq 3$, desconectamos os nove

caminhos, porém esses vértices não podem estar simultaneamente em K' , pois há arestas proibidas entre eles. Suponhamos que nenhum desses três vértices estejam em K' . Desse modo, três caminhos só poderiam ser desfeitos incluindo os vértices b_{ja} . Porém, para desfazer os outros seis caminhos, deveríamos retirar todos os vértices t_{ja} e t'_{ja} , pois a presença dos vértices b_{ja} em K' impede que os vértices b'_{ja} estejam em K' também. Com isso, temos que exatamente um vértice c_{ja} deve estar em K' .

Sem perda de generalidade, vamos assumir que o vértice $c_{j1} \in K'$. Isso elimina três caminhos (Figura 3.9). Na Figura 3.9, o vértice $c_1 \in K'$ aparece

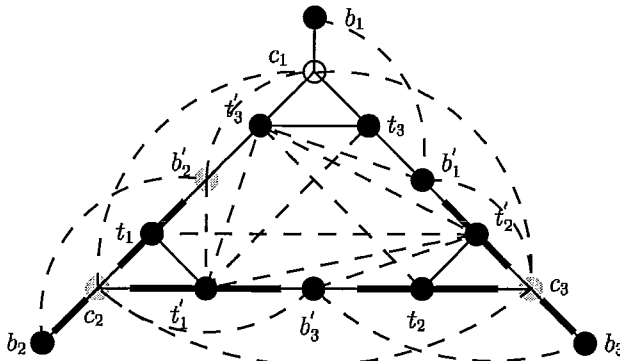


Figura 3.9: O vértice $c_{j1} \in K'$

em branco, e os vértices b'_3, c_2 e $c_3 \notin K'$ aparecem em cinza. É necessário incluir os vértices b_{j2} e b_{j3} para eliminar dois caminhos (Figura 3.10). Como os vértices b'_{j2} e b'_{j3} não podem estar em K' , pois $b_{j2} \in K'$ e $b_{j3} \in K'$, também devemos incluir os vértices t_{j1}, t'_{j1} e t_{j2} em K' , eliminando mais três caminhos (Figura 3.11). Resta apenas um caminho, a única maneira de eliminá-lo é incluindo o vértice b'_{j1} em K' , pois os outros vértices possuem arestas proibidas para vértices em K' (Figura 3.12). Se fossem os vértices c_{j2} ou c_{j3} que estivessem em K' , o conjunto K' poderia ser obtido de modo análogo, de acordo com Figura 3.13. Podemos construir o conjunto K' completamente

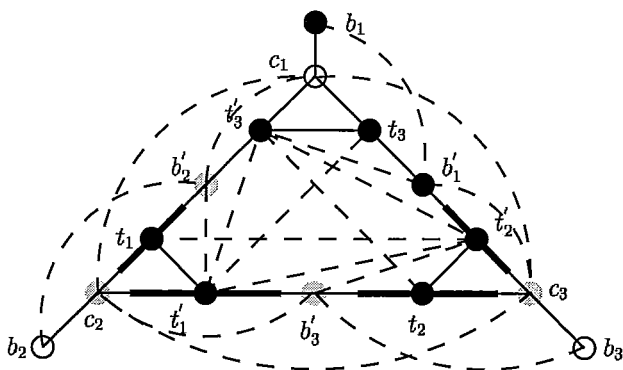


Figura 3.10: Os vértices b_{j2} e $b_{j3} \in K'$

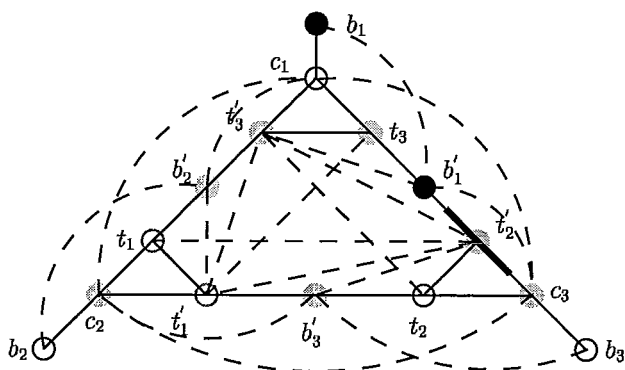


Figura 3.11: Os vértices t_{j1} , t'_{j1} e $t_{j2} \in K'$

escolhendo para cada i , exatamente um dentre os vértices x_i ou x'_i de cada subgrafo literal X_i , e para cada j , exatamente um dentre os vértices c_{j1} , c_{j2} ou c_{j3} de cada subgrafo cláusula C_j para estar em K' .

Apresentamos na Figura 3.14 um diagrama representando o grafo G' com um conjunto de vértices K' satisfazendo às propriedades: K' não possui aresta proibida e $G_1 \setminus K'$ é desconexo.

Lema 3.6 *O conjunto de vértices K' existe se e somente se $K' = K$ para alguma atribuição verdade.*

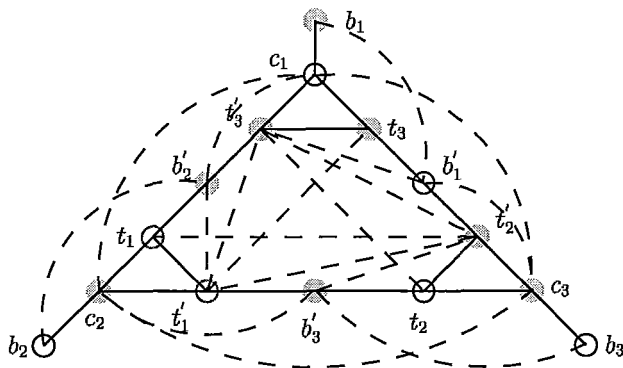


Figura 3.12: O vértice $b'_{j1} \in K'$

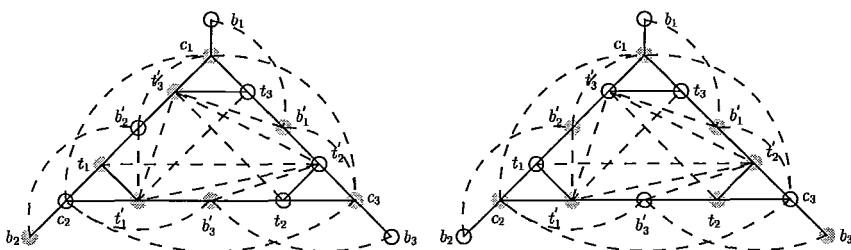


Figura 3.13: Os vértices c_{j2} e $c_{j3} \in K'$

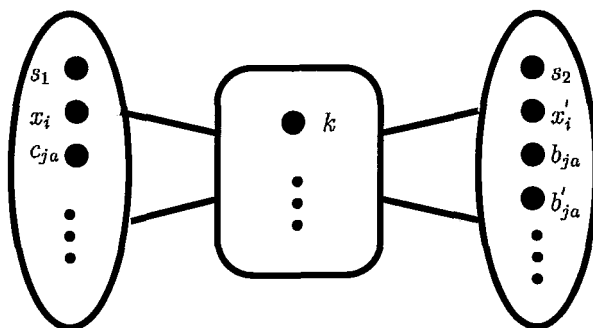


Figura 3.14: Diagrama representando os componentes separando s_1 de s_2

Prova. Primeiramente, precisamos mostrar que K' satisfaz a definição de K , assim como K satisfaz a definição de K' . Sabemos, pela Proposição 3.4, que o vértice $k \in K'$, e por definição de K , $k \in K$ também. Sabemos, pela

Proposição 3.3, que os vértices $s_1, s_2 \notin K'$, e por definição de K , $s_1, s_2 \notin K$ também. Na construção de K' , apenas um vértice c_{ja} de cada subgrafo C_j , e exatamente um dos vértices x_i e x'_i estão em K' . Fato idêntico ocorre na construção de K , devido a atribuição verdade. Comparando as Figuras 3.3 e 3.12 (nas quais temos em branco os vértices de K e de K' que estão em subgrafo cláusula), podemos ver que, de acordo com o vértice c_{ja} que estiver em K' ou em K , os vértices que completam os conjuntos K' e K , são os mesmos. Não existem mais outros vértices a serem adicionados nos dois conjuntos, e podemos notar que existe uma correspondência 1-1 entre os conjuntos K e os conjuntos K' .

Observamos também que a definição de K está de acordo com a definição de K' devido a uma atribuição verdade. Precisamos mostrar, finalmente, que K' também define uma atribuição verdade. Podemos construir uma atribuição verdade da seguinte forma: a variável x_i é verdadeira se e somente se $x_i \notin K'$. Caso essa atribuição verdade não exista, ocorre pelo menos uma dessas possibilidades:

(i) a existência de uma cláusula sem variável verdadeira;

(ii) a existência de uma cláusula com mais de uma variável verdadeira.

Pela construção de K' , exatamente um vértice c_{ja} de cada subgrafo cláusula C_j está em K' . Sem perda de generalidade, assumimos que este vértice é c_{11} de um subgrafo cláusula C_1 . Assim sendo $c_{11} \in K'$. Se a possibilidade (i) ocorrer com a cláusula C_1 , temos que sua variável associada x_1 é falsa. Logo, o vértice $x_1 \in K'$, porém, pela construção de G' , existe uma aresta proibida entre c_{11} e x_1 , assim K' não pode existir. Se a possibilidade (ii) ocorrer com a cláusula C_1 , temos que pelo menos uma outra variável x_2 é verdadeira associada ao vértice c_{12} de C_1 . Como a variável x_2 é verdadeira,

então o vértice $x'_2 \in K'$. Como, pela definição de K' , se $c_{11} \in K'$ então b_{12} também está, e, pela construção de G' , temos uma aresta proibida entre b_{12} e x'_2 , eles não podem estar ao mesmo tempo em K' . Isso contraria a definição de K' . Com isso temos que o conjunto K' só existe se houver uma atribuição verdade. ■

Teorema 3.7 *Existe uma atribuição satisfatível para o problema 1-EM-3 3SAT (sem literais negativos) se e somente se houver um grafo sanduíche $G(V, E)$ satisfazendo o PROBLEMA DO CORTE-CLIQUE SANDUÍCHE.*

Prova. Pela Proposição 3.1, mostramos que o subconjunto de vértices K de G' , formado a partir de uma atribuição verdade da instância do problema 1-EM-3 3SAT (sem literais negativos), é uma clique em G_2 . E pelo Lema 3.6, mostramos que K é um corte em G_1 . Logo, havendo uma atribuição satisfatível, podemos contruir um grafo sanduíche $G(V, E)$ que possua K como *corte-clique*, acrescentando ao grafo G_1 as arestas opcionais entre os vértices de K .

Reciprocamente, dado um *corte-clique* K , para um grafo sanduíche $G(V, E)$, temos que não existe aresta proibida entre vértices de K , e que $G_1 \setminus K$ é desconexo. Logo, pelo Lema 3.6, K corresponde a uma atribuição verdade. ■

Teorema 3.8 *CORTE-CLIQUE SANDUÍCHE é NP – Completo.*

Prova. O problema CORTE-CLIQUE SANDUÍCHE está em NP: um certificado é um grafo sanduíche G junto com o algoritmo polinomial de reconhecimento de *corte-clique* [12]. Pelo Teorema 3.7, temos que existe uma atribuição satisfatível para o problema 1-EM-3 3SAT (sem literais negativos)

se e somente se houver um grafo sanduíche $G(V, E)$ satisfazendo o CORTE-CLIQUE SANDUÍCHE. Mas sabemos que 1-EM-3 3SAT (sem literais negativos) é *NP - Completo* [11]. Assim temos que CORTE-CLIQUE SANDUÍCHE também é *NP-completo*. ■

3.3 Exemplo

Mostraremos a seguir, dois exemplos para o problema do CORTE-CLIQUE SANDUÍCHE.

No primeiro exemplo, construímos o grafo G' a partir de uma entrada para 1-EM-3 3SAT (sem literais negativos). Sejam as cláusulas (x_1, x_2, x_3) e (x_1, x_2, x_4) . Dadas essas cláusulas, construímos G' de acordo com a Figura 3.15. Note que, nesta figura, não estão representadas todas as arestas forçadas e proibidas de G' , mas apenas as arestas forçadas que correspondem às que aparecem nas Figuras 3.1 e 3.2, e as arestas proibidas específicas para a entrada de 1-EM-3 3SAT.

Mostraremos que, dada uma atribuição verdade para a entrada de 1-EM-3 3SAT, temos um *corte-clique*. Vemos que fazendo $x_1 = x_2 = F$ e $x_3 = x_4 = V$, temos uma atribuição verdade para a entrada dada. Podemos construir K de acordo com as instruções na página 21. Na Figura 3.16, onde os vértices de K aparecem em branco, podemos ver que K elimina todos os caminhos entre s_1 e s_2 . Na Figura 3.17, onde estão representadas, entre outras, as arestas proibidas específicas para a entrada, vemos que não existem arestas proibidas entre elementos de K .

Seja, agora, uma outra entrada para 1-EM-3 3SAT (sem literais negativos) tal como segue: (x_1, x_2, x_3) , (x_1, x_2, x_4) , (x_1, x_3, x_4) e (x_2, x_3, x_4) . O grafo G' correspondente a essa entrada está representado na Figura 3.18. Note que esse conjunto de cláusulas é insatisfável para 1-EM-3 3SAT, assim,

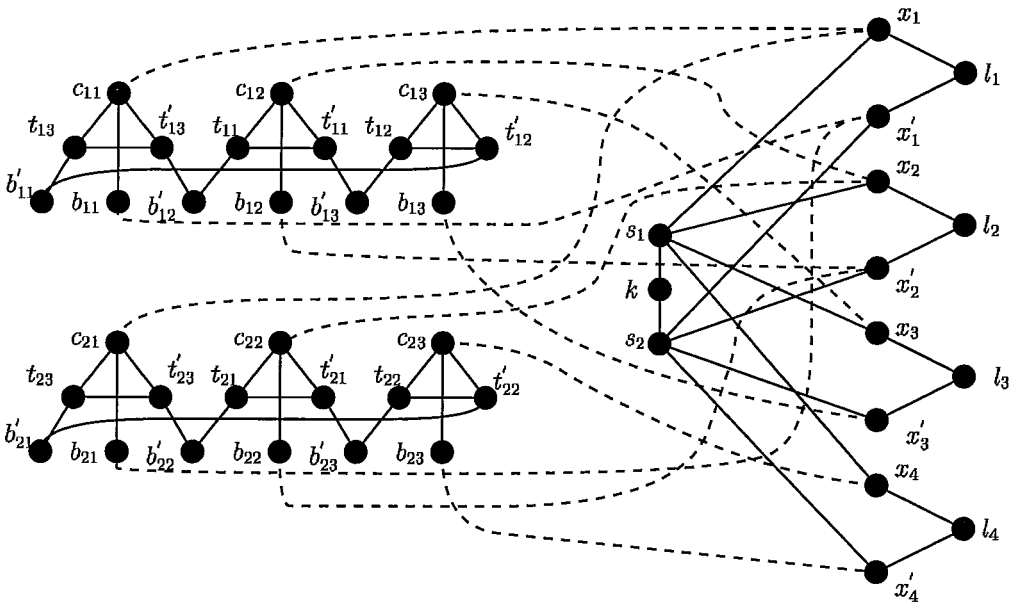


Figura 3.15: Grafo G'

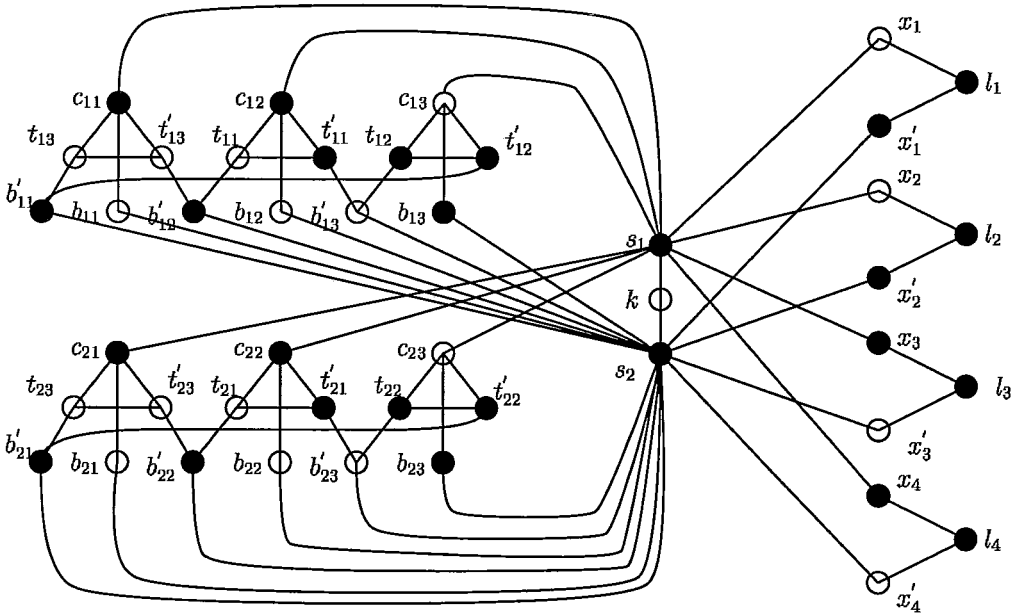


Figura 3.16: Exemplo 1 - Arestas Forçadas

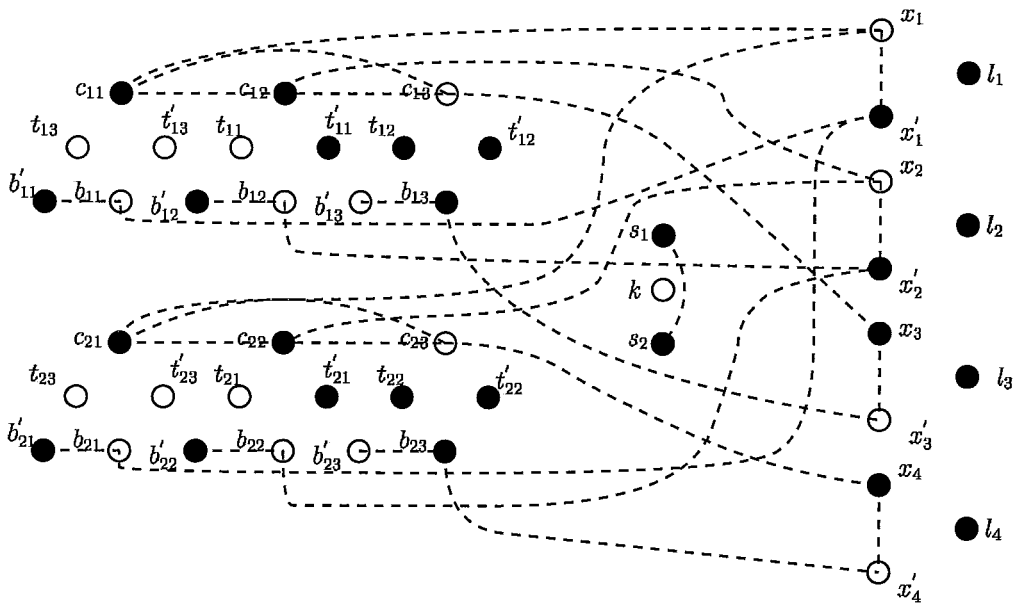


Figura 3.17: Exemplo 1 - Arestas Proibidas

de acordo com o Teorema 3.7, não podemos encontrar um *corte-clique*. De fato, dada qualquer atribuição para o conjunto de cláusulas, ao tentarmos construir K , temos um subgrafo cláusula C_j com todos vértice $c_{ja} \notin K$ ou algum subgrafo C_j com mais de um vértice $c_{ja} \in K$. Na primeira situação, temos que K não é corte, pois o subgrafo C_j garante um caminho entre s_1 e s_2 . Na segunda situação, K não é clique, pois existem arestas proibidas entre os vértices c_{ja} de C_j .

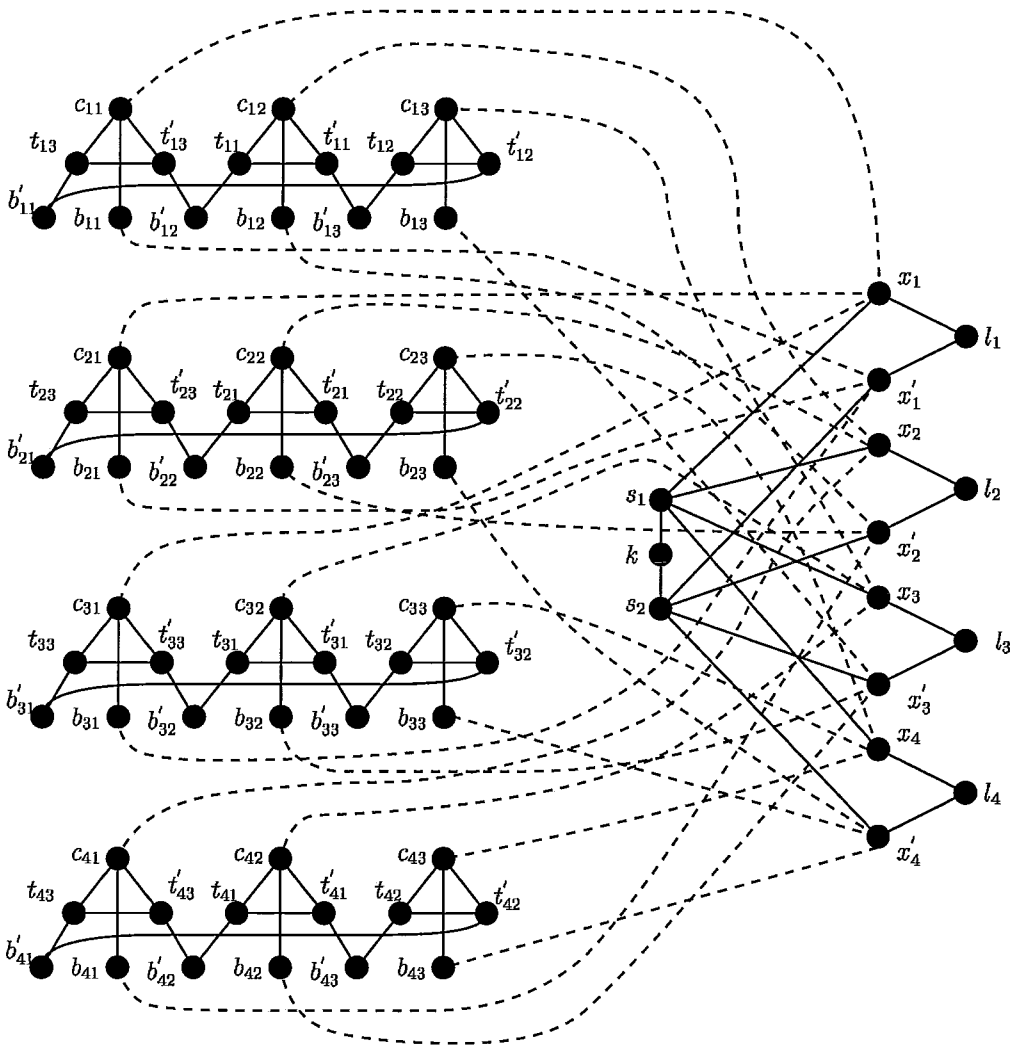


Figura 3.18: O grafo não possui *corte-clique*

Capítulo 4

Conclusões

Neste trabalho apresentamos resultados novos sobre problemas sanduíche. No Capítulo 1, mostramos a descrição dos problemas sanduíche, assim como introduzimos os problemas tratados nos capítulos seguintes.

No Capítulo 2, apresentamos um algoritmo para o problema do corte-estrela sanduíche. Calculamos a complexidade desse algoritmo, mostrando a sua polinomialidade de $O(n^3)$. Mostramos também um exemplo de execução de algoritmo enfatizando suas partes mais interessantes.

No Capítulo 3, apresentamos uma análise do problema sanduíche para corte-clique, mostrando sua relação com o problema NP -completo de reconhecimento de corte estável em um grafo. Essa relação nos levou pensar que o problema de corte-clique sanduíche seria também NP -completo e nos fez procurar por uma prova. Essa prova foi encontrada fazendo a redução do problema 1-EM-3 3SAT (sem literais negativos) para o problema sanduíche para corte-clique. Mostramos também um exemplo, onde dada uma entrada para o problema de satisfabilidade, geramos uma entrada para o problema sanduíche, fornecendo o resultado de ambos, comparando-os de acordo com a prova apresentada.

É interessante notar como dois problemas polinômias semelhantes não

estão mais na mesma classe de complexidade em suas versões sanduíche. Um corte-clique é também um corte-estrela e as complexidades dos algoritmos para suas versões de reconhecimento são bastante próximas.

4.1 Trabalhos futuros

Como trabalhos posteriores a este, podemos enumerar os seguintes:

- Reduzir a complexidade do algoritmo de corte-estrela sanduíche;
- Encontrar classes de grafos para as quais o problema do corte-clique sanduíche seja polinomial;
- Estudar problemas sanduíches para outras propriedades relacionadas a decomposição de grafos;
- Em relação ao artigo de Golumbic *et al.* onde o problema sanduíche foi apresentado [8], estudar três classes de grafos que ainda não têm o seu problema sanduíche classificado: grafos perfeitos, grafos fortemente cordais e grafos bipartidos cordais.

Referências Bibliográficas

- [1] BRANDSTÄDT, A., DRAGAN, F. F., LEE, V. B., AND SZYMCZAK, T. On stable cutsets in graphs. *Discrete Applied Mathematics*, 105 (2000), pp. 39–50.
- [2] CERIOLI, M. R., EVERETT, H., FIGUEIREDO, C. M. H., AND KLEIN, S. The homogeneous set sandwich problem. *Information Processing Letters*, 67 (1998), pp. 31–35.
- [3] CHVÁTAL, V. Star-cutsets and perfects graphs. *Journal of Combinatorial Theory Series B* 39, 3 (1985), pp. 189–199.
- [4] DANTAS, S. *Partições em Grafos: Caracterização, Algoritmos e Complexidade*. Tese de doutorado, COPPE/UFRJ, 2002.
- [5] DANTAS, S., DE FIGUEIREDO, C. M. H., AND FARIA, L. On the complexity of (k, l) -graph sandwich problems. *Proceedings of WG 2002 Lecture Notes in Computer Science 2573* (2002), pp. 92–101.
- [6] DE FIGUEIREDO, C. M. H., KLEIN, S., AND VUŠKOVIĆ, K. The graph sandwich problem for 1-join composition is NP-complete. *Discrete Applied Mathematics*, 121 (2002), pp. 73–82.
- [7] DE SÁ, V. G. P. *O Problema Sanduíche para Conjuntos Homogêneos em Grafos*. Dissertação de mestrado, COPPE/UFRJ, 2003.

- [8] GOLUMBIC, M. C., KAPLAN, H., AND SHAMIR, R. Graph sandwich problems. *Journal of Algorithms*, 19 (1995), pp. 449–473.
- [9] KLEIN, S., AND DE FIGUEIREDO, C. M. H. The NP-completeness of multi-partite cutset testing. *Congressus Numerantium 119* (1996), pp. 217–222.
- [10] KRATSCH, D., AND SPINRAD, J. Between $O(nm)$ and $O(n^\alpha)$. *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2003), pp. 158–167.
- [11] SCHAEFER, T. J. The complexity of the satisfiability problem. *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC)* (1978), pp. 216–226.
- [12] WHITESIDES, S. H. An algorithm for finding clique cutsets. *Information Processing Letters* 12, 1 (1981), pp. 31–32.