

Empacotamento de subgrafos em grafos

Gordana Manić

TESE APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DE GRAU DE DOUTOR
EM
CIÊNCIAS

Área de Concentração: Ciência da Computação

Orientadora: Profa. Dra. Yoshiko Wakabayashi

Durante a elaboração deste trabalho a autora recebeu apoio financeiro da CAPES

São Paulo, 8 de agosto de 2006

Empacotamento de subgrafos em grafos

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por Gordana Manić e aprovada pela Comissão Julgadora.

São Paulo, 30 de março de 2007

Banca Examinadora:

Profa. Dra. Yoshiko Wakabayashi (orientadora) – IME/USP

Prof. Dr. José Coelho de Pina Júnior – IME/USP

Prof. Dr. Flávio Keidi Miyazawa – IC/UNICAMP

Prof. Dr. Luerbio Faria – DMAT/UERJ

Prof. Dr. Marcelo Henriques de Carvalho – CCET/UFMS

À minha mãe, ao meu irmão
e à memória de meu pai
com afeto e estima.

Agradecimentos

À minha orientadora, professora Yoshiko Wakabayashi, pela sua excelente orientação, paciência e dedicação a este trabalho. Aos demais professores do grupo de Combinatória, por toda ajuda na minha formação acadêmica.

Aos meus pais e irmão, pelo incentivo incondicional em todas as minhas iniciativas, depositando em mim a confiança necessária para enfrentar qualquer dificuldade, e me dando o prazer de compartilhar com eles a alegria das minhas conquistas (*Mojim roditeljima i bratu, za podrsku i neophodno poverenje u svim mojim inicijativama. Njima dugujem za sve postignute uspehe*).

Ao professor Fernando Cortez Sica de Ouro Preto, ao Goroljub e Monica Milošević e a senhora Ivanka Ajdarić de Salvador que me incentivaram a ingressar no IME-USP, pelo apoio que me deram na ocasião, num dos momentos críticos. Ao senhor Dragiša Pavličić que me ajudou assim que cheguei à cidade de São Paulo.

Ao Cardonha, pelas conversas, apoio e amizade. Ao Frédéric Chataigner, pelas dicas e ajuda em diversas partes da minha tese, e ao Marcel Kenji pela leitura, correções e auxílio durante a redação deste texto.

Aos amigos desde os tempos do mestrado, Orlando Lee, Mário Leston, Pil e Renato de Carmo. A Maya pelos passeios, e aos demais amigos da salinha “vip”: Domingos Soares, Domingos Jr., Marcelo, Ellen, Luna, Cris, Fabrício, Trevizan e Ju, que juntos tornaram o meu dia-a-dia mais agradável. Ao Maurício, Geraldine, Thomas, Lilian e à galera da “salinha do café”.

À Nevena e ao Igor, Joca, Milan Lalić, Marijana e Pedja, Ana e Ivan, pela amizade e bate-papos indispensáveis em sérvio.

À Kémelli e à Claudete pela amizade, simpatia e pelos bons tempos que passamos juntos.

Aos amigos da minha cidade natal, Novi Sad, que mesmo estando longe são sempre uma constante fonte de energia e amizade: Borislava Ilić, Neda Divjaković e Ana Grubor.

Aos amigos Llane Alexis, Rino Carvalho, Maria de Souza e à moçada de Salvador, pelas férias inesquecíveis.

Resumo

Dentre os problemas clássicos da área de otimização combinatória distinguem-se os chamados problemas de empacotamento, que variam de acordo com os objetos a serem empacotados e as funções a serem otimizadas. Quando os objetos de interesse são grafos, vários desses problemas podem ser classificados como problemas de \mathcal{F} -empacotamento, onde \mathcal{F} é uma família de grafos. Estes problemas são assim definidos. Dado um grafo G , encontrar em G um subgrafo H que seja uma união de grafos disjuntos nos vértices (ou nas arestas), cada um dos quais é isomorfo a algum grafo da família \mathcal{F} , e tal que H tenha o maior número possível de vértices (ou arestas). Temos assim várias variantes, dependendo se consideramos disjunção nos vértices ou nas arestas, e se queremos maximizar o número de vértices ou arestas de H .

Um problema bem conhecido desse tipo é o problema do emparelhamento máximo, que corresponde ao caso em que $\mathcal{F} = \{K_2\}$ e H deve consistir de subgrafos disjuntos nos vértices. Nesta tese investigamos os casos $\mathcal{F} = \{K_3\}$ e $\mathcal{F} = \{K_2, K_3\}$, com ênfase no projeto de algoritmos de aproximação e resultados de inaproximabilidade.

No caso $\mathcal{F} = \{K_3\}$, estudamos o problema em que H deve ser uma união de triângulos disjuntos nos vértices, denotado por VTP, e o problema em que H deve ser uma união de triângulos disjuntos nas arestas e o objetivo é maximizar o número de arestas de H , denotado por ETP. Ambos os problemas são NP-difíceis. A melhor razão de aproximação conhecida para os dois problemas é $3/2 + \epsilon$, obtida por Hurkens e Schrijver em 1989. Apresentamos melhoras na razão de aproximação para casos restritos dos problemas VTP e ETP que são sabidamente APX-difíceis: apresentamos um algoritmo de aproximação para o VTP em grafos de grau máximo 4 com razão um pouco menor que 1,2, e para o ETP em grafos de grau máximo 5 com razão $4/3$. Apresentamos ainda um algoritmo linear exato para o VTP em grafos de intervalos unitários.

No caso em que $\mathcal{F} = \{K_2, K_3\}$, estudamos o problema em que os subgrafos de H devem ser disjuntos nos vértices e H deve ter o maior número possível de arestas. Provamos que este problema é APX-difícil mesmo em grafos de grau máximo 4. Ademais, apresentamos uma $(3/2 + \epsilon)$ -aproximação para grafos arbitrários e uma 1,4 -aproximação para grafos de grau máximo 4.

Palavras-chaves: algoritmos de aproximação. algoritmos polinomiais, APX-difícil, empacotamento de arestas, empacotamento de triângulos, grafos de intervalos unitários, grau limitado.

Abstract

A classical problem in graph theory is the problem of finding a maximum matching in a graph. This problem can be seen as an \mathcal{F} -packing problem, where \mathcal{F} is a family of graphs. This problem is defined as follows. Given a graph G , find a subgraph H in G that is a union of vertex-disjoint (or edge-disjoint) graphs, each of which is isomorphic to a graph in \mathcal{F} , and such that H has a maximum possible number of vertices (or edges). We may have several variants of this problem, according to the restrictions on H (vertex or edge disjoint version) and on the objective function.

In this thesis we investigate the cases in which $\mathcal{F} = \{K_3\}$ and $\mathcal{F} = \{K_2, K_3\}$, seeking approximation algorithms and inapproximability results.

When $\mathcal{F} = \{K_3\}$, we study the vertex-disjoint version, denoted here VTP, and the edge-disjoint version with the objective of maximizing the number of edges of H , denoted ETP. Both problems are NP-hard. The algorithm with the best approximation ratio known so far for these problems has ratio $3/2 + \varepsilon$, a result obtained by Hurkens and Schrijver in 1989. We present improvements on the approximation ratio for restricted cases of VTP and ETP that are known to be APX-hard: we give an approximation algorithm for VTP on graphs with maximum degree 4 with ratio slightly less than 1.2, and for ETP on graphs with maximum degree 5 with ratio $4/3$. We also present an exact linear-time algorithm for VTP on the class of unit interval graphs.

When $\mathcal{F} = \{K_2, K_3\}$, we study the vertex-disjoint version with the objective of maximizing the number of edges of H . We prove that this problem is APX-hard even on graphs with maximum degree 4. Furthermore, we present a $(3/2 + \varepsilon)$ -approximation algorithm for arbitrary graphs, and a 1.4-approximation algorithm for graphs with maximum degree 4.

Keywords: approximation algorithm, APX-hardness, edge packing, graph, low degree, polynomial algorithm, triangle packing, unit interval graph

Sumário

1	Introdução	1
1.1	Empacotamento de subgrafos em grafos	2
2	Preliminares	5
2.1	Teoria dos Grafos	5
2.2	Conjuntos e problemas de empacotamento	6
2.3	Classes de problemas, algoritmos e complexidade	7
3	Empacotamento de triângulos disjuntos nos vértices	9
3.1	Introdução	9
3.2	Grafos arbitrários	10
3.2.1	Algoritmo de Hurkens e Schrijver	10
3.3	Grafos com grau máximo 4	15
3.3.1	Introdução	15
3.3.2	Algoritmo de aproximação $VT4_k$	17
3.3.3	Desempenho do algoritmo $VT4_k$	22
3.3.4	Complexidade computacional do algoritmo $VT4_k$	32
3.3.5	Observações finais	33
3.4	Grafos de intervalos unitários	35
3.4.1	Introdução	35
3.4.2	Algoritmo exato	35
3.4.3	Observações finais	36
4	Empacotamento de triângulos disjuntos nas arestas	38
4.1	Introdução	38
4.2	Grafos arbitrários	38
4.3	Grafos com grau máximo 5	39
4.3.1	Algoritmo de aproximação $ET5$	39
4.3.2	Desempenho do algoritmo $ET5$	39
4.3.3	Complexidade computacional do algoritmo $ET5$	41

5	Empacotamento de arestas e triângulos	42
5.1	Introdução	42
5.2	Resultados de inaproximabilidade	43
5.3	Algoritmo de aproximação	50
5.4	Observações finais	53
6	Considerações finais	56
	Referências Bibliográficas	57
	Índice Remissivo	60

Capítulo 1

Introdução

Muitos problemas de otimização combinatória de grande importância e vasta aplicação prática são reconhecidamente NP-difíceis. Para esses problemas, os algoritmos conhecidos que encontram soluções ótimas consomem tempo exponencial, o que na prática é inviável. Em função do grande número e da importância desses problemas, diversas abordagens têm sido desenvolvidas na tentativa de se obter uma solução de compromisso. Dentre elas, as mais frequentes são: algoritmos pseudo-polinomiais, algoritmos probabilísticos, e algoritmos de aproximação.

Nosso interesse concentra-se na área de algoritmos de aproximação. As principais características dos algoritmos de aproximação é ser eficiente (polinomial) e ter garantia de desempenho. Essa garantia é uma medida da proximidade da solução encontrada pelo algoritmo em relação a uma solução ótima do problema. De acordo com a qualidade dessa medida, definem-se várias classes de algoritmos de aproximação. Nessa área, o interesse é projetar algoritmos de aproximação e também encontrar limitantes de aproximabilidade (resultados indicando que não é possível encontrar algoritmos com garantia de desempenho melhor do que certos limitantes, a menos que $P = NP$). Essa área teve um intenso desenvolvimento nas duas últimas décadas, em particular na década de 90, quando o famoso Teorema PCP foi provado por Arora et al. [1, 2, 3], fornecendo uma nova caracterização da classe de problemas NP.

Já na década de 60, o conceito de algoritmo de aproximação estava implícito em resultados obtidos por Graham [14] (sobre escalonamento em máquinas paralelas) e Erdős [13] (sobre grafos bipartidos). Mais tarde, Johnson [24] formalizou os conceitos de razão de aproximação e algoritmo de aproximação. Na década de 90 esta área passou a receber um tratamento mais sistemático. Surgiram técnicas aplicáveis a toda uma gama de problemas: métodos baseados em programação linear (método primal, dual e primal-dual), métodos probabilísticos, métodos baseados em programação semidefinida, etc. Na mesma época apareceram resultados de inaproximabilidade, mostrando que certos problemas podem ser aproximados, mas não tão bem como seria desejável (a menos que $P = NP$). Recentemente, diversos livros têm sido publicados sobre o assunto [4, 20, 29], mostrando seu reconhecimento como uma disciplina importante.

1.1 Empacotamento de subgrafos em grafos

Um problema clássico da área de otimização combinatória é o problema de encontrar, num dado grafo, um emparelhamento (coleção de arestas disjuntas) máximo. Uma generalização natural deste problema consiste em substituir *arestas disjuntas* por *subgrafos disjuntos*, sendo estes pré-estabelecidos, dados através de uma família \mathcal{F} de grafos. Este problema é conhecido como \mathcal{F} -empacotamento, definido da seguinte forma. Dado um grafo $G = (V, E)$ encontre um subgrafo H em G , tal que H seja uma união de grafos (disjuntos nos vértices ou arestas), cada um dos quais é isomorfo a algum grafo da família \mathcal{F} , e tal que H tenha o maior número possível de vértices ou arestas.

Notamos que há várias variantes desse problema: pode-se considerar grafos disjuntos nos vértices ou nas arestas, e a função objetivo pode ser maximizar o número de vértices ou o número de arestas que pertencem a H (no caso mais geral, pode-se ter pesos associados aos vértices e/ou às arestas). Não existem muitas famílias \mathcal{F} para as quais as propriedades de aproximabilidade e/ou inaproximabilidade dos correspondentes problemas de \mathcal{F} -empacotamento são completamente entendidas. A maioria dos resultados obtidos trata da complexidade computacional desses problemas.

Surpreendentemente, o problema do \mathcal{F} -empacotamento pode ser resolvido em tempo polinomial para algumas famílias não-triviais \mathcal{F} , e muitos resultados importantes sobre emparelhamentos podem ser generalizados para esses casos. Listamos, em seguida, algumas das famílias \mathcal{F} estudadas até agora para o problema de encontrar um subgrafo H que seja uma união de *grafos disjuntos nos vértices* e que tenha o *maior número possível de vértices*.

- \mathcal{F} consiste de um único grafo com pelo menos 3 vértices em alguma componente conexo: o problema é NP-difícil. Este resultado segue do resultado provado em [18].
- \mathcal{F} consiste de uma aresta e um conjunto de grafos hipo-emparelháveis (hypomatchable): o problema está em P [19, 12]. (Um grafo G é hipo-emparelhável se $G - v$ tem um emparelhamento perfeito para todo $v \in V_G$.)
- \mathcal{F} consiste de estrelas S_i , $1 \leq i \leq r$, onde r é um inteiro fixo: está em P [17].

Questões relativas à aproximabilidade de problemas de \mathcal{F} -empacotamento (que são NP-difíceis) foram investigadas para algumas poucas famílias \mathcal{F} . Dentre essas famílias, destaca-se o caso em que \mathcal{F} consiste de um único triângulo (uma clique de tamanho 3). Este caso, aparentemente simples, é um problema clássico que tem sido objeto de alguns estudos, mas que ainda carece de resultados mais satisfatórios. Por exemplo, para grafos arbitrários, seria interessante responder a questão da existência ou não de algoritmos com garantia de aproximação melhor do que as já conhecidas; para certas classes especiais de grafos, exibir algoritmos polinomiais, ou melhorar razões conhecidas e provar limitantes de aproximação. Este problema e um outro correlato são os assuntos centrais da nossa pesquisa. Mencionamos a seguir esses problemas.

Empacotamento Máximo de Triângulos Disjuntos nos Vértices (VTP):
 dado um grafo simples G , encontrar uma coleção máxima de triângulos em G dois a dois disjuntos nos vértices.

Empacotamento Máximo de Triângulos Disjuntos nas Arestas (ETP): dado um grafo simples G , encontrar uma coleção máxima de triângulos em G dois a dois disjuntos nas arestas.

$K_2 + K_3$ -empacotamento: é o problema do \mathcal{F} -empacotamento no caso em que $\mathcal{F} = \{K_2, K_3\}$, e o objetivo é encontrar $H \subseteq G$ que seja uma união de grafos disjuntos nos vértices, cada um dos quais é isomorfo a um grafo de \mathcal{F} , e tal que H tenha o maior número possível de arestas.

O problema VTP surge em problemas de escalonamento e problemas de 3-agrupamentos de pessoas: dado um conjunto de pessoas e afinidades entre elas, encontrar grupos disjuntos de 3 membros cada, de modo que as pessoas num mesmo grupo sejam compatíveis e o número de pessoas isoladas seja mínimo [11]. O problema ETP tem aplicações em biologia computacional [7]. Os problemas VTP e ETP são NP-difíceis [25, 21].

Um dos resultados mais relevantes para os problemas VTP e ETP foram obtidos por Hurkens e Schrijver [23] em 1989. O resultado geral de Hurkens e Schrijver [23] sobre empacotamento máximo de k -conjuntos implica que um algoritmo simples de busca local é uma $(\frac{3}{2} + \epsilon)$ -aproximação para o VTP e o ETP. Esta é a melhor razão de aproximação que se conhece para os dois problemas (para grafos arbitrários).

Para um dado inteiro $k \geq 3$, denotamos por VTP- k (resp. ETP- k) o problema VTP (resp. ETP) restrito a grafos com grau máximo k . Em 2002, Caprara e Rizzi [7] exibiram algoritmos polinomiais para os problemas VTP-3 e ETP-4. Eles mostraram ainda que o problema VTP-4 (veja também [6]) e ETP-5 são APX-difíceis (isto é, não admitem um esquema de aproximação polinomial, a menos que $P = NP$).

Considerando que, em termos de grau, instâncias com grau máximo 4 para o VTP (resp. 5 para o ETP) constituem um caso difícil que está no limite entre instâncias difíceis e fáceis [7], investimos um tempo no estudo dos problemas VTP-4 e ETP-5, na tentativa de encontrar um resultado positivo de aproximabilidade ou um limitante inferior para a razão de aproximação. Conseguimos [27] melhorar a razão de aproximação para esses casos APX-difíceis: obtivemos um algoritmo de aproximação para o problema VTP-4 cuja razão de aproximação é $3 - \frac{\sqrt{13}}{2} + \epsilon$, e um algoritmo para o ETP-5 com razão de aproximação $\frac{4}{3}$.

Para um dado grafo G , denotamos por $\mathcal{T}_V(G)$ (resp. $\mathcal{T}_E(G)$) a coleção dos conjuntos dos vértices (resp. arestas) de todos os triângulos em G . É fácil ver que o problema VTP num grafo G pode ser reduzido ao problema de encontrar um subconjunto de vértices independente de cardinalidade máxima (problema MIS) no grafo de interseção de $\mathcal{T}_V(G)$. Claramente, esta é uma AP-redução: um algoritmo de ρ -aproximação para o problema MIS leva a um algoritmo de ρ -aproximação para o VTP. Nosso algoritmo para o problema VTP-4 se baseia nesse fato. Ele aplica no grafo de entrada G algumas reduções, que preservam a razão de aproximação, para obter um grafo G' no qual todo triângulo intersecta no máximo 3 outros triângulos. Em seguida, no grafo de interseção de $\mathcal{T}_V(G')$, aplica o algoritmo de $\rho(k)$ -aproximação de Chlebík e Chlebíková [10], onde $\rho(k) = 3 - \frac{\sqrt{13}}{2} + \frac{13 - \sqrt{13}}{52k}$ e k é um parâmetro inteiro fixo, para o problema MIS em grafos de grau máximo 3. Observamos que $\rho(4)$ é um pouco menor que 1,25; e $\rho(k)$ é um pouco menor que 1,2 para $k > 65$.

Já o algoritmo que desenvolvemos para o problema ETP-5 é baseado numa busca local. A idéia desse algoritmo consiste em verificar se existe um certo subgrafo H (chamado grafo

de *Hajós*) em G . Se existe, o algoritmo adiciona (de maneira definitiva) alguns triângulos de H ao conjunto \mathcal{A}^* a ser devolvido pelo algoritmo, retira E_H do grafo G e repete o processo. Caso contrário, ele aplica o algoritmo de busca local de Hurkens e Schrijver mencionado anteriormente.

Obtivemos também um algoritmo linear para o problema VTP em grafos de intervalos unitários. Um *grafo de intervalos* é um grafo de interseção de um conjunto finito de intervalos em \mathbb{R} . Quando todos esses intervalos são de mesmo tamanho, um tal grafo é chamado *grafo de intervalos unitários*. Nosso algoritmo é guloso e faz uso da caracterização de tais grafos obtida por Looges e Olariu [26]: um grafo $G = (V, E)$ é um grafo de intervalos unitários se e somente se existe uma ordem linear $<$ dos vértices de G tal que, para cada escolha de vértices u, v, w temos que “se $u < v < w$ e $uw \in E$ então $uv, vw \in E$.”

O problema do $K_2 + K_3$ -empacotamento, na versão aqui considerada (maximizar o número de arestas de H), tem sido pouco investigada. Encontramos apenas um artigo de Halldórsson [16] sobre cobertura de k -conjuntos que fornece uma $3/2$ -aproximação para esse problema. Observamos que quando o objetivo é maximizar o número de vértices de H então o problema correspondente pode ser resolvido em tempo polinomial [19]. Já o problema de nosso interesse é NP-difícil.

Provamos que o problema do $K_2 + K_3$ -empacotamento é APX-difícil para grafos com grau máximo 5. Para isso, mostramos uma L-redução do problema VTP restrito a grafos de grau máximo 4. Mostramos também que o $K_2 + K_3$ -empacotamento é APX-difícil mesmo para grafos de grau máximo 4 e onde toda aresta pertence a algum triângulo. A L-redução é do problema MAX2SAT3. Ela é mais elaborada, e é baseada na idéia de Caprara e Rizzi [7] para mostrar que o problema VTP-4 é APX-completo.

Além disso, mostramos uma $(1 + \frac{1}{3}\rho)$ -aproximação para o $K_2 + K_3$ -empacotamento, onde ρ é uma razão de aproximação para o problema VTP. Nosso algoritmo inicialmente usa uma rotina para encontrar um número máximo de triângulos disjuntos nos vértices, e depois complementa a solução com um emparelhamento máximo no grafo obtido removendo-se os vértices dos triângulos que a rotina encontrou. Usando como rotina o algoritmo de Hurkens e Schrijver [23] para o VTP, temos que a razão de aproximação do nosso algoritmo é $\frac{3}{2} + \varepsilon$. Ressaltamos ainda que o algoritmo de aproximação para o VTP-4 que obtivemos (com razão um pouco menor que 1,2) resulta em um algoritmo de aproximação com razão 1,4 para o problema do $K_3 + K_2$ -empacotamento em grafos de grau máximo 4.

Apresentamos aqui uma $3/2$ -aproximação para o problema do $K_2 + K_3$ -empacotamento proposta recentemente por Yuster.

Capítulo 2

Preliminares

Neste capítulo introduzimos alguns conceitos e estabelecemos a notação que utilizamos nos próximos capítulos.

2.1 Teoria dos Grafos

Um *grafo* G é um par (V_G, E_G) de conjuntos finitos, onde E_G é um conjunto de pares não-ordenados de elementos distintos de V_G . Quando G estiver subentendido, omitimos o índice G . Por exemplo, em vez de V_G e E_G , escrevemos simplesmente V e E . Os elementos de V_G são chamados *vértices*, e os elementos de E_G de *arestas*. Por brevidade, denotamos uma aresta $\{u, v\}$ por uv ou vu . Se $\alpha = uv$, dizemos que α *incide em* u e em v e que u e v são os *extremos* de α . Também dizemos que u e v são *vizinhos* ou *adjacentes* e que u e v são *ligados* por α . Duas arestas distintas são ditas *adjacentes* se elas têm um extremo em comum. Observamos que o objeto que definimos como grafo é normalmente chamado de grafo simples.

Um grafo G é *vazio* se $V_G = E_G = \emptyset$. A *ordem* de um grafo G , denotada por n_G , é o número de seus vértices. O *grau* de um vértice v de um grafo G é o número de arestas incidentes em v e será denotado por $d_G(v)$. Um vértice de um grafo é *isolado* se seu grau é zero. O *grau máximo* é o número $\Delta(G) := \max\{d_G(v) : v \in V_G\}$.

Um conjunto de vértices ou arestas dois a dois não-adjacentes é chamado *independente*. No caso de vértices, um conjunto independente é também chamado *estável*. Um conjunto independente de arestas de um grafo é chamado de *emparelhamento*.

Se todos os vértices de um grafo G de ordem n são dois a dois adjacentes dizemos que G é *completo* e o denotamos por K_n ; o grafo K_3 é chamado *triângulo*.

Um grafo H é um *subgrafo* de um grafo G se $V_H \subseteq V_G$ e $E_H \subseteq E_G$; escrevemos $H \subseteq G$. A *união* de dois grafos G e H , denotada por $G \cup H$, é o grafo $(V_G \cup V_H, E_G \cup E_H)$. A *interseção* de dois grafos G e H , denotada por $G \cap H$, é o grafo $(V_G \cap V_H, E_G \cap E_H)$. Dois grafos G e H se *intersectam* se $G \cap H$ é um grafo não-vazio.

Se U é um subconjunto de V_G , a *vizinhança* de U , denotada por $\Gamma_G(U)$, é o conjunto dos vértices de $V_G \setminus U$ adjacentes a pelo menos um vértice de U .

Se $U \subseteq V_G$, então o subgrafo de G *induzido* por U , denotado por $G[U]$, é o subgrafo de G cujo conjunto de vértices é U e cujo conjunto de arestas é o conjunto das arestas de G que têm ambos os extremos em U . Denotamos por $G - U$ o grafo $G[V_G \setminus U]$; para o caso em que

$U = \{v\}$, abreviamos $G - \{v\}$ por $G - v$. Similarmente para $U \subseteq E_G$.

Um grafo é *bipartido* se V_G admite uma partição em duas classes de modo que cada aresta tem extremos em classes distintas; uma tal partição é chamada *bipartição* do grafo. Um grafo *bipartido completo* é um grafo bipartido no qual todo par de vértices de classes distintas é uma aresta. Denotamos por $K_{m,n}$ o grafo completo com bipartição (V_1, V_2) tal que $|V_1| = m$ e $|V_2| = n$.

Um *caminho* é um grafo P com $V_P = \{v_1, \dots, v_k\}$ não-vazio e $E_P = \{v_1v_2, v_2v_3, \dots, v_{k-1}v_k\}$. Podemos representar um tal caminho por $v_1v_2 \cdots v_k$. Se $P = v_1v_2 \cdots v_k$ é um caminho com $k \geq 3$, então $(V_P, E_P \cup \{v_kv_1\})$ é chamado de *circuito*. Uma *corda* de um circuito é uma aresta que liga dois vértices não consecutivos deste circuito. O *comprimento* de um caminho é o número de suas arestas; definimos analogamente o comprimento de um circuito.

Um grafo é *conexo* se, para todo par de vértices distintos u e v , contém um caminho com extremos u e v . Os subgrafos conexos maximais de um grafo são chamados *componentes*. Uma *clique* em um grafo é um conjunto de vértices dois a dois adjacentes.

O *grafo de interseção* de uma coleção de conjuntos \mathcal{T} é o grafo H com $V_H := \mathcal{T}$ e $XY \in E_H \Leftrightarrow X \cap Y \neq \emptyset$. Um *grafo de intervalos* é um grafo de interseção de um conjunto finito de intervalos em \mathbb{R} . Quando todos esses intervalos são de mesmo tamanho, um tal grafo é chamado *grafo de intervalos unitários* (ou *grafo indiferença*). Um grafo é *cordal* se todo circuito de tamanho maior que 3 tem uma corda. Um grafo é *livre de k -garras* (*k -claw free*) se ele não contém nenhum subgrafo completo bipartido $K_{1,k}$ como subgrafo induzido. Um *cografo* é um grafo que não tem caminho de comprimento 4 induzido. Um grafo é *split* se o seu conjunto de vértices pode ser particionado em um conjunto independente e uma clique.

Para um dado número real $\lambda > 0$, denote por C_λ um conjunto finito de círculos de mesmo raio no plano, onde a distância entre os centros de quaisquer dois círculos é pelo menos λ . Um grafo *disco-unitário de λ -precisão* é um grafo cujos vértices correspondem aos círculos de algum conjunto C_λ (como definimos), dois de seus vértices são adjacentes se e somente se os círculos correspondentes se intersectam.

O problema do *Conjunto Independente Máximo* (MIS) é definido como segue. Dado um grafo G , encontre um subconjunto de vértices independente de cardinalidade máxima em G .

2.2 Conjuntos e problemas de empacotamento

Para uma dada família \mathcal{F} de grafos o problema do \mathcal{F} -empacotamento é definido da seguinte forma. Dado um grafo G , encontre um subgrafo H em G , tal que H seja uma união de grafos (disjuntos nos vértices ou arestas), cada um dos quais é isomorfo a algum grafo da família \mathcal{F} , e tal que H tenha o maior número possível de vértices ou arestas.

Estudaremos o problema do \mathcal{F} -empacotamento no caso em que $\mathcal{F} = \{K_3\}$, e o objetivo é encontrar um grafo $H \subseteq G$ que seja uma união de *grafos disjuntos nos vértices* (resp. *arestas*), sendo cada um deles isomorfo a K_3 , tal que H tenha o *maior número possível de arestas*. Esses problemas também são conhecidos pela seguinte denominação:

Problema do *Empacotamento Máximo de Triângulos Disjuntos nos Vértices* (VTP): dado um grafo G , encontrar uma coleção máxima de triângulos em G dois a dois disjuntos nos vértices. Problema do *Empacotamento Máximo de Triângulos Disjuntos nas Arestas* (ETP): dado um grafo G , encontrar uma coleção máxima de triângulos em G dois a dois disjuntos nas

arestas. Para um dado grafo G , denotamos por $\mathcal{T}_V(G)$ (resp. $\mathcal{T}_E(G)$) a coleção dos conjuntos dos vértices (resp. arestas) de todos os triângulos em G . Para simplificar, vamos nos referir a uma coleção de triângulos disjuntos nos vértices (resp. arestas) de G como um *empacotamento* de $\mathcal{T}_V(G)$ (resp. $\mathcal{T}_E(G)$). Para um dado inteiro $k \geq 3$, denotamos por VTP- k (resp. ETP- k) o problema VTP (resp. ETP) restrito a grafos com grau máximo k .

Uma redução natural que podemos fazer nas instâncias dos problemas VTP e ETP consiste em retirar os vértices isolados e as arestas que não pertencem a nenhum triângulo. Portanto, no estudo dos problemas de empacotamento de triângulos, restringimos nossa atenção a grafos nos quais toda aresta pertence a algum triângulo; tais grafos serão chamados *irredundantes*.

Denotamos por \mathcal{T}_G a coleção de todos os triângulos em G . Um triângulo cujos vértices são u, v e w é denotado por $[u, v, w]$. O grau de $T \in \mathcal{T}_G$, denotado por $d_G(T)$, é o número de triângulos em G diferentes de T que intersectam T .

Um empacotamento de $\{K_2, K_3\}$ em um grafo G é um subgrafo H em G tal que H é uma união de grafos disjuntos nos vértices, cada um dos quais é isomorfo a K_2 ou K_3 . Se \mathcal{A} é um empacotamento de $\{K_2, K_3\}$ em um grafo G , então o valor de \mathcal{A} é o número de arestas de G que \mathcal{A} cobre, e o conjunto $\mathcal{T}_{\mathcal{A}}$ (resp. $\mathcal{E}_{\mathcal{A}}$) é a coleção de todos os triângulos (resp. arestas) em \mathcal{A} . O problema do $K_2 + K_3$ -empacotamento é o problema de encontrar num grafo G um empacotamento de $\{K_2, K_3\}$ de valor máximo.

Um k -conjunto é um conjunto com exatamente k elementos. Dada uma coleção \mathcal{F} de conjuntos, dizemos que uma subcoleção \mathcal{T} de \mathcal{F} cobre um conjunto finito E se todo elemento de E pertence a algum conjunto de \mathcal{T} . Neste caso, dizemos também que \mathcal{T} é uma *cobertura* de E . O problema da *cobertura mínima por k -conjuntos* (k -SC) consiste no seguinte: dados um conjunto finito E e uma coleção finita \mathcal{F} de subconjuntos de E tamanho no máximo k , encontrar uma cobertura \mathcal{T} de E de cardinalidade mínima.

Seja V um conjunto finito de elementos e $E_1, \dots, E_l \subseteq V$. Dizemos que a coleção E_1, \dots, E_l tem um *sistema de representantes distintos* (SRD) se existem elementos e_1, \dots, e_l dois a dois distintos tais que $e_i \in E_i$, para todo $i = 1, \dots, l$. Uma coleção E_1, \dots, E_l tem um t -SRD se qualquer coleção de no máximo t conjuntos entre E_1, \dots, E_l tem um SRD. Note que pelo teorema de Hall, as seguintes duas afirmações são equivalentes:

- Uma coleção E_1, \dots, E_l tem um t -SRD
- Para todo $p \leq t$, quaisquer p conjuntos entre E_1, \dots, E_l cobrem pelo menos p elementos de V .

2.3 Classes de problemas, algoritmos e complexidade

Um *algoritmo de ρ -aproximação* para um problema de maximização Π é um algoritmo polinomial tel que, para toda instância I do problema, devolve uma solução S cujo valor, $\text{val}_{\Pi}(I, S)$, é pelo menos $\frac{1}{\rho} \text{opt}_{\Pi}(I)$, onde $\text{opt}_{\Pi}(I)$ é o valor de uma solução ótima de I . (o parâmetro ρ pode ser um número maior ou igual a 1, ou uma função do tamanho da instância). Se um tal algoritmo existe, Π pertence à classe de problemas APX. Dizemos que ρ é uma *razão de aproximação* ou *garantia* do algoritmo. Um *esquema de aproximação polinomial* (PTAS) para Π é uma família de algoritmos $\{A_{\varepsilon} : \varepsilon \in (0, 1)\}$ tal que para cada ε , A_{ε} é um algoritmo de $\frac{1}{1-\varepsilon}$ -aproximação para Π .

Denotamos por $\text{Sol}(I)$ o conjunto das soluções viáveis de uma instância I . Uma AP-redução de um problema de otimização Π_1 para um problema de otimização Π_2 é um terno (f, g, β) em que f e g são algoritmos e β é um número racional positivo tais que:

- f recebe um número racional positivo δ e uma instância I de Π_1 , e devolve uma instância $f(\delta, I)$ de Π_2 ,
- g recebe um número racional positivo δ , uma instância I de Π_1 e um elemento S_2 em $\text{Sol}(f(\delta, I))$, e devolve $g(\delta, I, S_2)$ em $\text{Sol}(I)$,
- para todo número racional positivo δ , os algoritmos $f(\delta, \cdot)$ e $g(\delta, \cdot, \cdot)$ são polinomiais,
- para toda instância I de Π_1 , todo número racional positivo δ , e todo S_2 em $\text{Sol}(f(\delta, I))$, vale que se

$$(1 - \delta)\text{opt}(f(\delta, I)) \leq \text{val}(f(\delta, I), S_2) \leq (1 + \delta)\text{opt}(f(\delta, I)),$$

$$\text{então } (1 - \beta\delta)\text{opt}(I) \leq \text{val}(I, g(\delta, I, S_2)) \leq (1 + \beta\delta)\text{opt}(I).$$

Se existe uma AP-redução de um problema de otimização Π_1 a um problema de otimização Π_2 , então dizemos que Π_1 é AP-reduzível a Π_2 . Um problema de otimização Π é APX-difícil se todo problema em APX é AP-reduzível a Π , e APX-completo se é APX-difícil e está em APX.

Uma L-redução de um problema de otimização Π_1 para um problema de otimização Π_2 é uma quádrupla (f, g, α, β) que consiste de dois algoritmos polinomiais f e g e duas constantes positivas α e β satisfazendo as seguintes condições:

- para toda instância I_1 de Π_1 , temos que $f(I_1)$ é uma instância de Π_2 ,
- dada uma instância I_1 de Π_1 e qualquer solução viável S de $f(I_1)$, temos que $g(I_1, S)$ é uma solução viável para a instância I_1 de Π_1 ,
- $|\text{opt}_{\Pi_2}(f(I_1))| \leq \alpha|\text{opt}_{\Pi_1}(I_1)|$, para toda instância I_1 de Π_1 ,
- $|\text{opt}_{\Pi_1}(I_1) - \text{val}_{\Pi_1}(I_1, g(I_1, S))| \leq \beta|\text{opt}_{\Pi_2}(f(I_1)) - \text{val}_{\Pi_2}(f(I_1), S)|$, para toda instância I_1 de Π_1 e toda solução viável S de $f(I_1)$.

Usamos a notação $\Pi_1 \leq_L \Pi_2$ para indicar que existe uma L-redução de Π_1 para Π_2 . Se $\Pi_1 \leq_L \Pi_2$ e $\Pi_2 \in \text{APX}$, então $\Pi_1 \in \text{APX}$. Mais precisamente, a existência de uma $(\frac{1}{1-\epsilon})$ -aproximação para Π_2 implica a existência de uma $(\frac{1}{1-\alpha\beta\epsilon})$ -aproximação para Π_1 . Em se tratando de problemas em APX, a existência de uma L-redução implica na existência de uma AP-redução. Mais precisamente, se $\Pi_1 \leq_L \Pi_2$ e Π_1 está em APX, então Π_1 é AP-reduzível a Π_2 . Assim, para mostrar que um problema Π_2 é APX-difícil basta mostrar que existe um problema APX-completo Π_1 tal que $\Pi_1 \leq_L \Pi_2$ (veja em [4]).

Capítulo 3

Empacotamento de triângulos disjuntos nos vértices

3.1 Introdução

O problema de Empacotamento Máximo de Triângulos Disjuntos nos Vértices (VTP) é NP-difícil [25, 21]. Temos interesse em obter algoritmos de aproximação para esse problema, resultados de inaproximabilidade, ou obter classes/famílias de instâncias para as quais este problema é polinomial.

Hurkens e Schrijver [23] apresentaram um algoritmo para o problema do empacotamento máximo de k -conjuntos, $k \geq 2$: dada uma família \mathcal{F} de k -conjuntos, encontre um empacotamento máximo de \mathcal{F} . Trata-se de um algoritmo simples de busca local. No caso particular em que $k = 3$, tal algoritmo é uma $(\frac{3}{2} + \varepsilon)$ -aproximação para o VTP. A razão de aproximação obtida por Hurkens e Schrijver é a melhor razão de aproximação que se conhece para o VTP (para grafos arbitrários).

Em 1994, Baker [5] provou que o problema VTP restrito a grafos planares admite um esquema de aproximação polinomial (PTAS). Conforme [7], este resultado pode ser estendido para o ETP. Os dois problemas admitem tal esquema também para grafos disco-unitários de λ -precisão [22]. O problema VTP é NP-difícil mesmo quando restrito a grafos cordais e é polinomial para as classes de grafos *split* e cografos [15].

Para um dado inteiro $k \geq 3$, denotamos por VTP- k (resp. ETP- k) o problema VTP (resp. ETP) restrito a grafos com grau máximo k . Em 2002, Caprara e Rizzi [7] mostraram que o VTP-3 está em P, e que o VTP-4 (veja também [6]) é APX-difícil, isto é, não admite um esquema de aproximação polinomial a menos que $P = NP$. Chlebík e Chlebíková [9] mostraram que $95/94$ é um limitante inferior para a razão de aproximação para o problema VTP-4.

Apresentamos agora um resultado simples que usaremos na análise da complexidade de nossos algoritmos.

Lema 3.1 *O número de triângulos em um grafo G de grau máximo Δ é no máximo $\frac{\Delta(\Delta-1)n_G}{6}$.*

Demonstração. Claramente, toda aresta de G pertence a no máximo $\Delta - 1$ triângulos. Então $|E|(\Delta - 1)$ contabiliza o número máximo possível de triângulos em G , onde cada triângulo é

contado 3 vezes (uma vez para cada uma de suas arestas). Como $|E| \leq \frac{\Delta n_G}{2}$, temos que

$$|\mathcal{T}_G| \leq \frac{|E|(\Delta - 1)}{3} \leq \frac{\Delta(\Delta - 1)n_G}{6}.$$

□

Neste capítulo apresentamos inicialmente o resultado obtido por Hurkens e Schrijver [23] em 1989 para o problema VTP. Em seguida, apresentamos um algoritmo de aproximação para o VTP-4 cuja razão de aproximação é $3 - \frac{\sqrt{13}}{2} + \varepsilon$, e um algoritmo exato para o VTP em grafos de intervalos unitários.

Neste capítulo restringimos nossa atenção aos grafos irredundantes. Quando escrevemos $G - U$ (para $U \subseteq V_G$ ou $U \subseteq E_G$), estamos supondo que os vértices isolados e as arestas que não pertencem a nenhum triângulo foram removidos também.

3.2 Grafos arbitrários

Observamos que um algoritmo guloso simples que retorna uma coleção maximal de triângulos disjuntos nos vértices é uma 3-aproximação para o problema VTP. Isto porque todo triângulo da uma solução obtida pelo algoritmo guloso intersecta no máximo 3 triângulos de uma solução ótima.

3.2.1 Algoritmo de Hurkens e Schrijver

Um dos resultados mais relevantes para o problema VTP segue de um resultado mais geral (sobre empacotamento máximo de k -conjuntos), obtido por Hurkens e Schrijver [23] em 1989. Descrevemos a seguir esse resultado. Dados um número inteiro fixo t e uma coleção \mathcal{T} de k -conjuntos, considere o seguinte algoritmo.

Algoritmo HS(\mathcal{T}, t):

Dada uma coleção $\mathcal{C} \subseteq \mathcal{T}$ já construída, de conjuntos dois a dois disjuntos, verifique se existem $p \leq t$ conjuntos dois a dois disjuntos em $\mathcal{T} \setminus \mathcal{C}$ que intersectam no máximo $p - 1$ conjuntos da coleção \mathcal{C} . Se existir, troque os conjuntos, obtendo uma coleção maior \mathcal{C} , e repita o processo. Caso contrário, devolva \mathcal{C} (que chamamos de t -ótimo).

A troca acima consiste em incluir em \mathcal{C} os novos p conjuntos de $\mathcal{T} \setminus \mathcal{C}$ e remover no máximo $p - 1$ conjuntos que os intersectam.

Note que o algoritmo HS(\mathcal{T}, t) se baseia numa busca local e encontra uma coleção de conjuntos em \mathcal{T} dois a dois disjuntos. Claramente, este algoritmo resolve o VTP. Basta tomar $k = 3$ e $\mathcal{T} = \mathcal{T}_V(G)$, onde G é o grafo de entrada. Lembramos que $\mathcal{T}_V(G)$ denota a coleção dos conjuntos dos vértices de todos os triângulos em G . O resultado de Hurkens e Schrijver [23] sobre empacotamento máximo de k -conjuntos implica que o algoritmo HS(\mathcal{T}, t) é uma $(\frac{3}{2} + \varepsilon)$ -aproximação para o VTP, onde ε é inversamente proporcional a t (conforme t cresce, ε tende a zero). Eles exibiram ainda, para cada t , instâncias para as quais o algoritmo acima produz soluções que atingem tal razão em relação ao empacotamento máximo de triângulos disjuntos

nos vértices. A razão de aproximação obtida por Hurkens e Schrijver é a melhor razão de aproximação que se conhece para o VTP (para grafos arbitrários).

Apresentamos em seguida o resultado geral de Hurkens e Schrijver [23] sobre empacotamento máximo de conjuntos, derivando daí o resultado para o problema VTP.

Teorema 3.2 *Seja Y um conjunto com k elementos e E_1, E_2, \dots, E_l subconjuntos de Y . Se todo elemento de Y está contido em no máximo 3 dos conjuntos E_1, E_2, \dots, E_l , e qualquer coleção de no máximo t conjuntos entre E_1, E_2, \dots, E_l tem um SRD, então*

$$\begin{aligned} \frac{l}{k} &\leq \frac{3 \cdot 2^r - 3}{2 \cdot 2^r - 3}, \text{ se } t = 2r - 1 \text{ e} \\ \frac{l}{k} &\leq \frac{3 \cdot 2^r - 2}{2 \cdot 2^r - 2}, \text{ se } t = 2r. \end{aligned} \quad (3.1)$$

Lembramos que a coleção E_1, \dots, E_l tem um sistema de representantes distintos se existem elementos e_1, \dots, e_l dois a dois distintos tais que $e_i \in E_i$, para todo $i = 1, \dots, l$. Para provar este teorema, ordenamos os conjuntos E_1, \dots, E_l de modo que

$$\begin{aligned} |E_1| &= \dots = |E_q| \geq 3 \\ |E_{q+1}| &= \dots = |E_h| = 2 \\ |E_{h+1}| &= \dots = |E_l| = 1. \end{aligned} \quad (3.2)$$

Definimos ainda o conjunto $W := \bigcup_{i=h+1}^l E_i$. Sejam T_1, \dots, T_k os elementos de Y . Para cada $i = 1, \dots, q$ seja X_i um conjunto de tamanho $|E_i| - 2$, disjunto de $E_1 \cup \dots \cup E_l$, e tal que $X_i \cap X_j = \emptyset$, se $i \neq j$. Seja $X_1 \cup \dots \cup X_q = \{y_1, \dots, y_s\}$. Considere agora a coleção de conjuntos

$$(E_1 \setminus W) \cup X_1, \dots, (E_q \setminus W) \cup X_q, E_{q+1} \setminus W, \dots, E_h \setminus W, \{y_1\}, \dots, \{y_s\}. \quad (3.3)$$

Para a prova do Teorema 3.2 precisaremos do seguinte lema.

Lema 3.3 *Para todo $t \geq 3$, se E_1, \dots, E_l tem um t -SRD, então a coleção 3.3 tem um $(t-2)$ -SRD.*

Demonstração. Suponha que a coleção (3.3) não tenha um $(t-2)$ -SRD. Portanto, existe uma coleção Π de conjuntos de (3.3) tal que $|\Pi| = p$, $p \leq t-2$ e $|\cup \Pi| \leq p-1$. Escolha Π de forma que p seja mínimo. Então,

$$|\cup \Pi| = p - 1. \quad (3.4)$$

De fato, suponha que $|\Pi| = p$ e $|\cup \Pi| \leq p-2$, e considere a coleção $\Pi' := \Pi \setminus \{S\}$, onde S é qualquer elemento de Π . Temos que $p' := |\Pi'| = p-1$ e $|\cup \Pi'| \leq p-2 = p' - 1$. Porém $p' < p$, o que contradiz a minimalidade de p .

Ainda, temos que

$$\text{cada elemento de } \cup \Pi \text{ é coberto por pelo menos dois conjuntos em } \Pi. \quad (3.5)$$

De fato, suponha que exista um elemento T de $\cup \Pi$ tal que T seja coberto por exatamente um conjunto S em Π . Considere $\Pi' := \Pi \setminus \{S\}$. Como T é coberto somente por S , temos que

$\cup \Pi' \subseteq \cup \Pi \setminus \{T\}$. Usando (3.4) temos que $p' := |\Pi'| = p-1$ e $|\cup \Pi'| \leq |\cup \Pi| - 1 = p-2 = p'-1$, o que contradiz a minimalidade de p .

Mostramos em seguida que de (3.5) segue a seguinte afirmação.

$$\text{Para cada } i = 1, \dots, q \text{ e } x \in X_i, \text{ temos que } \{x\} \in \Pi \Leftrightarrow (E_i \setminus W) \cup X_i \in \Pi. \quad (3.6)$$

Suponha que $\{x\} \in \Pi$. Claramente, $x \in \cup \Pi$. De (3.5) temos que x pertence a mais algum conjunto em Π . Como todo X_i é disjunto de $E_1 \cup \dots \cup E_l$ e $X_i \cap X_j = \emptyset$ para $i \neq j$, temos que os únicos conjuntos da família (3.3) que contêm x são $\{x\}$ e $(E_i \setminus W) \cup X_i$. Logo, $(E_i \setminus W) \cup X_i \in \Pi$. Analogamente, $(E_i \setminus W) \cup X_i \in \Pi$ implica que $\{x\} \in \Pi$.

Podemos supor, sem perda de generalidade, que

todos os conjuntos $(E_1 \setminus W) \cup X_1, \dots, (E_q \setminus W) \cup X_q, E_{q+1} \setminus W, \dots, E_h \setminus W$ pertencem a Π . (3.7)

De fato, suponha que exista j , $1 \leq j \leq h$, tal que $(E_j \setminus W) \cup X_j \notin \Pi$ (ou $E_j \setminus W \notin \Pi$). Considere a família $\mathcal{E}' := \{E_1, \dots, E_l\} \setminus \{E_j\}$. Como E_1, \dots, E_l tem um t -SRD, sabemos que quaisquer s ($s \leq t$) conjuntos entre E_1, \dots, E_l cobrem pelo menos s elementos de V . Considere s conjuntos em \mathcal{E}' ($s \leq t$). Estes conjuntos pertencem à coleção E_1, \dots, E_l , portanto cobrem pelo menos s elementos de V . Ademais, como Π não foi mudado, temos que $|\Pi| = p$ e $|\cup \Pi| \leq p-1$. Logo, podemos remover E_j de E_1, \dots, E_l se $(E_j \setminus W) \cup X_j \notin \Pi$ (ou $E_j \setminus W \notin \Pi$).

Note que

$$\text{os conjuntos } E_{h+1}, \dots, E_l \text{ são dois a dois disjuntos.} \quad (3.8)$$

Suponha a existência de dois conjuntos E_{i_1}, E_{i_2} , $h+1 \leq i_1 < i_2 \leq l$, tais que $E_{i_1} = E_{i_2}$. Como $|E_{i_1}| = |E_{i_2}| = 1$, concluímos que a coleção E_1, \dots, E_l não tem um t -SRD (pois $t \geq 3$), o que é uma contradição.

Ainda, sem perda de generalidade, podemos supor que

$$(E_1 \cup \dots \cup E_h) \cap W = E_{h+1} \cup \dots \cup E_l. \quad (3.9)$$

De fato, como $W = \bigcup_{i=h+1}^l E_i$, temos que $(E_1 \cup \dots \cup E_h) \cap W \subseteq E_{h+1} \cup \dots \cup E_l$. Suponha agora que $E_{h+1} \cup \dots \cup E_l \not\subseteq (E_1 \cup \dots \cup E_h) \cap W$. Então, existe T_j , $1 \leq j \leq k$, tal que $T_j \in E_{h+1} \cup \dots \cup E_l = W$ e $T_j \notin (E_1 \cup \dots \cup E_h) \cap W$, ou seja, $T_j \notin E_1 \cup \dots \cup E_h$. De (3.8) temos que existe somente um conjunto E_i , $h+1 \leq i \leq l$, tal que $\{T_j\} = E_i$. Ainda, como $T_j \notin E_1 \cup \dots \cup E_h$, temos que T_j pertence a somente um conjunto de Π , o que contradiz (3.5).

Note que

$$s = |X_1 \cup \dots \cup X_q| = \sum_{i=1}^h (|E_i| - 2). \quad (3.10)$$

Além disso,

$$p = h + s. \quad (3.11)$$

De fato, por (3.7), todos os conjuntos $(E_1 \setminus W) \cup X_1, \dots, (E_q \setminus W) \cup X_q, E_{q+1} \setminus W, \dots, E_h \setminus W$ pertencem a Π . Logo, por (3.6) concluímos que $\{x\} \in \Pi$ para todo $x \in X_i$ e $i = 1, \dots, q$. Segue imediatamente que Π é toda a coleção (3.3), ou seja, $|\Pi| = p = h + s$.

Do fato de que Π é toda a coleção (3.3), e das afirmações (3.4) e (3.11) segue que

$$\left| \bigcup_{i=1}^h (E_i \setminus W) \right| = |\cup \Pi| - s = (p-1) - s = (p-s) - 1 = h-1. \quad (3.12)$$

Agora, de (3.9) temos que

$$\left| \bigcup_{i=1}^l (E_i \cap W) \right| = \left| \bigcup_{i=1}^h (E_i \cap W) \right| = |W|. \quad (3.13)$$

Ainda, de (3.8) segue que

$$|W| = l - h. \quad (3.14)$$

Usando (3.12), (3.13) e (3.14), temos que

$$\begin{aligned} \left| \bigcup_{i=1}^l E_i \right| &= \left| \bigcup_{i=1}^h (E_i \setminus W) \right| + \left| \bigcup_{i=1}^l (E_i \cap W) \right| \\ &= \left| \bigcup_{i=1}^h (E_i \setminus W) \right| + \left| \bigcup_{i=1}^h (E_i \cap W) \right| \\ &= (h-1) + |W| = h-1 + (l-h) = l-1. \end{aligned} \quad (3.15)$$

Ainda, por (3.5), temos que $\sum_{i=1}^h |E_i \setminus W| \geq 2 \left| \bigcup_{i=1}^h (E_i \setminus W) \right|$. Usando esta inequação, o fato de que $p \leq t-2$, e as afirmações (3.14), (3.9), (3.12), (3.10) e (3.11), temos que

$$\begin{aligned} l &= h + (l-h) \\ &= h + |E_{h+1} \cup \dots \cup E_l| \\ &= h + |(E_1 \cup \dots \cup E_h) \cap W| \\ &= h + \left| \bigcup_{i=1}^h (E_i \cap W) \right| \leq h + \sum_{i=1}^h |E_i \cap W| = h + \sum_{i=1}^h (|E_i| - |E_i \setminus W|) \\ &= h + \sum_{i=1}^h |E_i| - \sum_{i=1}^h |E_i \setminus W| \leq h + \sum_{i=1}^h |E_i| - 2 \left| \bigcup_{i=1}^h (E_i \setminus W) \right| \\ &= h + 2h + \sum_{i=1}^h |E_i - 2| - 2(h-1) \\ &= h + 2h + s - 2h + 2 \\ &= h + s + 2 = p + 2 \leq t. \end{aligned} \quad (3.16)$$

Então, de (3.15) e (3.16) temos que $\left| \bigcup_{i=1}^l E_i \right| = l-1$ e $l \leq t$, o que contradiz o fato de que a coleção E_1, \dots, E_l tem um t -SRD. \square

Apresentamos a seguir a prova do Teorema 3.2.

Demonstração. (do Teorema 3.2) A prova é por indução em t . Denotamos por $\varphi(t)$ o termo do lado direito das inequações em (3.1). Note que, como todo elemento de Y está

contido em no máximo três conjuntos de E_1, \dots, E_l , temos que

$$\sum_{i=1}^l |E_i| \leq 3k. \quad (3.17)$$

Seja $t = 1$. Como E_1, \dots, E_l tem um t -SRD, temos que todo E_i , $i = 1, \dots, l$ é não-vazio. Logo, $l \leq \sum_{i=1}^l |E_i|$. De (3.17) temos que $l \leq 3k$, ou seja, $l/k \leq 3 = \varphi(1)$.

Seja $t = 2$. Logo, qualquer coleção de no máximo 2 conjuntos entre E_1, \dots, E_l tem um SRD. Portanto, $E_i \neq \emptyset$, para $i = 1, \dots, l$. Ainda, a união de quaisquer 2 conjuntos entre E_1, \dots, E_l cobrem pelo menos 2 elementos de Y . Portanto, não existem 2 conjuntos E_{i_1}, E_{i_2} , $h+1 \leq i_1 < i_2 \leq l$, tais que $E_{i_1} = E_{i_2}$. Logo,

$$l - h \leq k. \quad (3.18)$$

Agora, de (3.2), (3.14) e (3.17) temos que

$$l + h = 2h + (l - h) \leq \sum_{i=1}^h |E_i| + \sum_{i=h+1}^l |E_i| = \sum_{i=1}^l |E_i| \leq 3k.$$

Usando a inequação acima e (3.18), obtemos $2l = (l - h) + (l + h) \leq k + 3k = 4k$. Logo, $l/k \leq 2 = \varphi(2)$.

Seja agora $t \geq 3$. Considere a coleção (3.3) de conjuntos sobre

$$Y' := \bigcup_{i=1}^{l'} E'_i.$$

Note que

$$l' = h + s \text{ e } k' := |Y'| = |Y| - |W| + s = k - |W| + s. \quad (3.19)$$

Pelo Lema 3.3, temos que $E'_1, \dots, E'_{l'}$ tem um $(t-2)$ -SRD. Claramente, todo elemento de Y' está em no máximo 3 dos conjuntos $E'_1, \dots, E'_{l'}$. Logo, pela hipótese de indução,

$$l' \leq \varphi(t-2)k'.$$

Usando a inequação acima e (3.19), concluímos que $h + s \leq \varphi(t-2)(k - |W| + s)$. Reescrevendo essa última desigualdade obtemos

$$\varphi(t-2)|W| + h - (\varphi(t-2) - 1)s \leq \varphi(t-2)k. \quad (3.20)$$

Agora, de (3.10) e (3.17) temos que

$$|W| + 2h + s = |W| + 2h + \sum_{i=1}^h (|E_i| - 2) = |W| + \sum_{i=1}^h |E_i| = \sum_{i=1}^l |E_i| \leq 3k. \quad (3.21)$$

Usando (3.14), temos que

$$\begin{aligned} l &= (l - h) + h = |W| + h \\ &= \frac{|W|(2\varphi(t-2) - 1)}{2\varphi(t-2) - 1} + \frac{h(2\varphi(t-2) - 1)}{2\varphi(t-2) - 1} = \frac{2|W|\varphi(t-2) - |W| + 2h\varphi(t-2) - h}{2\varphi(t-2) - 1} \quad (3.22) \\ &= \frac{\varphi(t-2)|W| + h - \varphi(t-2)s + s + \varphi(t-2)|W| - |W| + 2h\varphi(t-2) - 2h + s\varphi(t-2) - s}{2\varphi(t-2) - 1} \\ &= \frac{1}{2\varphi(t-2) - 1} (\varphi(t-2)|W| + h - (\varphi(t-2) - 1)s) + \frac{\varphi(t-2) - 1}{2\varphi(t-2) - 1} (|W| + 2h + s). \end{aligned}$$

Agora, (3.20), (3.21) e (3.22) implicam que

$$l \leq \frac{1}{2\varphi(t-2)-1}\varphi(t-2)k + \frac{\varphi(t-2)-1}{2\varphi(t-2)-1}3k = k \frac{4\varphi(t-2)-3}{2\varphi(t-2)-1}.$$

Logo,

$$\frac{l}{k} \leq \frac{4\varphi(t-2)-3}{2\varphi(t-2)-1}. \quad (3.23)$$

Para $t = 2r - 1$, usando (3.23) e a hipótese de indução, temos que

$$\frac{l}{k} \leq \frac{4 \frac{3 \cdot 2^{r-1} - 3}{2 \cdot 2^{r-1} - 3} - 3}{2 \frac{3 \cdot 2^{r-1} - 3}{2 \cdot 2^{r-1} - 3} - 1} = \frac{3 \cdot 2^r - 3}{2 \cdot 2^r - 3} = \varphi(t).$$

Para $t = 2r$, usando (3.23) e a hipótese de indução, temos que

$$\frac{l}{k} \leq \frac{4 \frac{3 \cdot 2^{r-1} - 2}{2 \cdot 2^{r-1} - 2} - 3}{2 \frac{3 \cdot 2^{r-1} - 2}{2 \cdot 2^{r-1} - 2} - 1} = \frac{3 \cdot 2^r - 2}{2 \cdot 2^r - 2} = \varphi(t).$$

□

Corolário 3.4 *Seja $G = (V, E)$ um grafo, $\mathcal{T}_V(G)$ a coleção dos conjuntos dos vértices de todos os triângulos em G , e t um inteiro positivo. O algoritmo $\text{HS}(\mathcal{T}_V(G), t)$ é uma $(\frac{3}{2} + \varepsilon)$ -aproximação para o problema VTP, onde ε é inversamente proporcional a t .*

Demonstração. Sejam T_1, \dots, T_k os triângulos devolvidos pelo algoritmo $\text{HS}(\mathcal{T}_V(G), t)$, e T'_1, \dots, T'_l um empacotamento máximo de $\mathcal{T}_V(G)$. Definimos $Y := \{T_1, \dots, T_k\}$. Para cada $i = 1, \dots, l$, definimos o conjunto $E_i := \{T_j \mid T_j \cap T'_i \neq \emptyset\}$. Observe que todo T_j , $j = 1, \dots, k$, está contido em no máximo três conjuntos de E_1, \dots, E_l , pois os triângulos T'_1, \dots, T'_l são dois a dois disjuntos nos vértices. Ainda, como a solução do algoritmo $\text{HS}(\mathcal{T}_V(G), t)$ é t -ótima, temos que a união de quaisquer p , $p \leq t$ triângulos de T'_1, \dots, T'_l intersecta pelo menos p triângulos de T_1, \dots, T_k . Portanto, qualquer coleção de no máximo t conjuntos entre E_1, \dots, E_l tem um SRD. Pelo Teorema 3.2 temos que

$$\begin{aligned} \frac{l}{k} &\leq \frac{3 \cdot 2^r - 3}{2 \cdot 2^r - 3}, \text{ se } t = 2r - 1 \\ \frac{l}{k} &\leq \frac{3 \cdot 2^r - 2}{2 \cdot 2^r - 2}, \text{ se } t = 2r. \end{aligned}$$

□

3.3 Empacotamento de triângulos disjuntos nos vértices em grafos com grau máximo 4

3.3.1 Introdução

Nesta seção restringimos nossa atenção aos grafos irredundantes de grau máximo 4.

Considerando que, em termos de grau, instâncias com grau máximo 4 para o problema VTP constituem um caso difícil que está no limite entre instâncias difíceis e fáceis [7], investimos um tempo no estudo do problema VTP-4, procurando obter um algoritmo com uma razão de aproximação melhor. Desenvolvemos um algoritmo de aproximação para o problema VTP-4 cuja razão de aproximação é $3 - \frac{\sqrt{13}}{2} + \varepsilon$.

É fácil ver que o problema VTP num grafo G pode ser reduzido ao problema de encontrar um subconjunto de vértices independentes de cardinalidade máxima (problema MIS) no grafo de interseção de $\mathcal{T}_V(G)$. Claramente, esta é uma AP-redução: um algoritmo de ρ -aproximação para o problema MIS leva a um algoritmo de ρ -aproximação para o VTP.

Nesta seção descrevemos um algoritmo de aproximação, chamado de VT4 $_k$, para o problema VTP em grafos irredundantes de grau máximo 4, onde k é um parâmetro inteiro fixo. Este algoritmo aplica no grafo de entrada G algumas reduções que preservam a razão de aproximação, e obtém um grafo G' no qual todo triângulo intersecta no máximo 3 outros triângulos. Em seguida, no grafo de interseção de $\mathcal{T}_V(G')$, aplica o algoritmo MIS3 $_k$ para o problema MIS em grafos de grau máximo 3. O algoritmo MIS3 $_k$, obtido por Chlebík e Chlebíková [10], é uma $\rho(k)$ -aproximação onde $\rho(k) = 3 - \frac{\sqrt{13}}{2} + \frac{13 - \sqrt{13}}{52k}$. Observamos que $\rho(4)$ é um pouco menor que 1,25; e $\rho(k)$ é um pouco menor que 1,2 para $k > 65$. Vale notar que o algoritmo MIS3 $_k$ é essencialmente o algoritmo de Berman e Fujito [6], mas a análise cuidadosa feita por Chlebík e Chlebíková em [10] melhora a razão ($\frac{6}{5} + \frac{1}{5k}$) obtida em [6].

O algoritmo MIS3 $_k$ aplica inicialmente reduções ao grafo de entrada G' que preservam a razão de aproximação, entre outros: *Branchy Reduction* - se existe um caminho $v_1v_2v_3v_4$ onde v_2 e v_3 têm grau 2, remove v_2 e v_3 do grafo e insere a aresta v_1v_4 , e *Small Commitment Reduction* - procura repetidamente um subconjunto I de $V_{G'}$ tal que $|I| \leq k$ e I é um conjunto independente máximo no subgrafo induzido pelos vértices em I e seus vizinhos. Se encontrado, I é adicionado ao conjunto que o algoritmo MIS3 $_k$ devolverá e os vértices em I e seus vizinhos são removidos do grafo G' .

O algoritmo MIS3 $_k$ é baseado em aumento local em duas direções. Um *p-aumento local* de um conjunto independente I adiciona p vértices e retira $p - 1$ vértices de I , de modo que o conjunto obtido é independente. Um conjunto independente é *p-ótimo* se ele não admite um *p-aumento local*.

Se I é um conjunto independente no grafo de entrada G' , o algoritmo MIS3 $_k$ iterativamente aplica $O(k \log |V_{G'}|)$ -aumentos em I . Quando I se torna $O(k \log |V_{G'}|)$ -ótimo, o algoritmo encontra um conjunto independente máximo I' no grafo induzido por $V_{G'} \setminus I$ (como o grau máximo de G' é 3, todo vértice no grafo $G'[V_{G'} \setminus I]$ tem grau no máximo 2, e portanto I' é fácil de encontrar). Se $|I'| > |I|$, o processo se repete com $|I'|$ no lugar de $|I|$. Caso contrário, o algoritmo devolve I .

O consumo de tempo para encontrar aumentos locais é $O(|V_{G'}|^{1+3k \log 3})$. Como tais aumentos são feitos no máximo $|V_{G'}|$ vezes, o tempo de execução do algoritmo MIS3 $_k(G')$ é $O(|V_{G'}|^{2+3k \log 3})$.

Em todas as figuras desta seção, cada vértice quadrado é um vértice comum a dois triângulos de G cuja união é uma borboleta. Um vértice x que é representado por um círculo é *saturado*, ou seja, nenhuma outra aresta é incidente a x , além das que já estão na figura.

Introduzimos agora alguns conceitos que utilizamos neste capítulo. Se dois triângulos T_1 e T_2 de G têm um único vértice em comum e não existe outro triângulo em G que intersecta

ambos T_1 e T_2 , dizemos que o subgrafo $T_1 \cup T_2$ é uma *borboleta* em G e denotamos por $v_{T_1 T_2}$ o único vértice comum a T_1 e T_2 . Uma coleção \mathcal{T} de triângulos disjuntos nos vértices em G é *localmente ótima* em G se $\{V_T: T \in \mathcal{T}\}$ é um empacotamento máximo da coleção $\{V_T: T \in \mathcal{T}_G, T \text{ intersecta um triângulo em } \mathcal{T}\}$. Quando $\mathcal{T} = \{T\}$, dizemos simplesmente que o triângulo T é localmente ótimo.

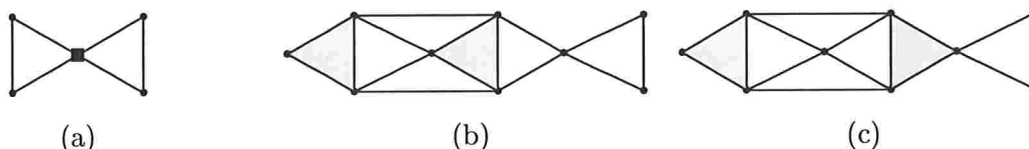


Figura 3.1: (a) Uma borboleta. (b) Uma coleção localmente ótima. (c) Uma coleção não localmente ótima.

3.3.2 Algoritmo de aproximação $VT4_k$

Antes de darmos uma descrição formal do algoritmo $VT4_k$, apresentamos as idéias básicas desse algoritmo. Em cada iteração do algoritmo $VT4_k$, uma coleção $\mathcal{T} \subseteq \mathcal{T}_G$, $|\mathcal{T}| \leq 2$, localmente ótima em G é repetidamente adicionada a \mathcal{A}^* (solução que o algoritmo devolve) e G é atualizado. Se G tem um triângulo T de grau maior que 3, o algoritmo encontra um certo subgrafo H que contém T e aplica uma redução apropriada. A definição de borboleta é crucial pois a redução é baseada no número de triângulos em H que formam uma borboleta com um triângulo fora de H (este número é no máximo 2, como veremos na seção 3.3.3). Também, vamos provar que, para uma coleção \mathcal{T} localmente ótima em G , a adição de \mathcal{T} à solução atual e a remoção de G dos vértices de todos os triângulos em \mathcal{T} preservam a razão de aproximação do algoritmo $VT4_k$. Ainda, provaremos que cada redução feita no subgrafo H preserva a razão de aproximação do algoritmo.

Descrevemos agora o algoritmo $VT4_k$ em mais detalhes. Em cada iteração do algoritmo $VT4_k$, uma coleção $\mathcal{T} \subseteq \mathcal{T}_G$, $|\mathcal{T}| \leq 2$, localmente ótima em G é repetidamente adicionada a \mathcal{A}^* (e G é atualizado) com o objetivo de eliminar instâncias especiais, ou seja, instâncias que têm uma coleção localmente ótima com no máximo 2 triângulos (algumas dessas instâncias especiais são mostradas na Figura 3.2). Fazendo isto, como vamos ver na seção 3.3.3, restam somente algumas poucas instâncias gerais que têm uma estrutura similar.

Se G ainda contém um triângulo T de grau maior que 3, o algoritmo encontra um subgrafo H definido como um subgrafo maximal conexo irredundante de G que contém T e não contém nenhuma borboleta. Explorando a estrutura do grafo G irredundante de grau máximo 4, e usando o fato de que G não contém nenhuma coleção $|\mathcal{T}|$, $|\mathcal{T}| \leq 2$, localmente ótima em G (caso contrário, ela seria adicionada à solução do algoritmo na primeira etapa), podemos provar que tal grafo H tem uma estrutura muito específica. Mais precisamente, o número de triângulos T' em \mathcal{T}_H para os quais existe um triângulo em $\mathcal{T}_G \setminus \mathcal{T}_H$ que forma uma borboleta com T' em G é no máximo 2. Além disso, H é isomorfo a um dos grafos da Figura 3.16. Ressaltamos porém que, o subgrafo H pode ser encontrado sem fazer uma busca exaustiva de subgrafos isomorfos aos da Figura 3.16.

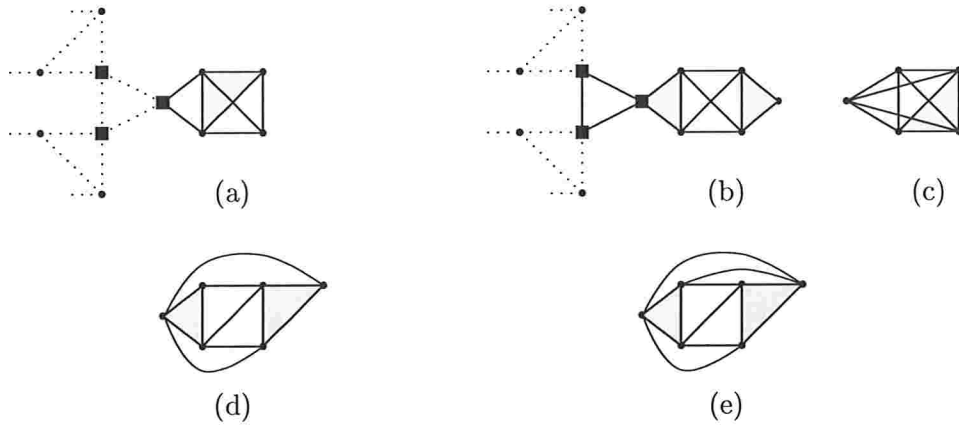


Figura 3.2: Exemplos das instâncias que têm uma coleção \mathcal{T} localmente ótima com no máximo dois triângulos (os triângulos marcados estão em \mathcal{T} , e cada vértice quadrado é comum a dois triângulos em G cuja união é uma borboleta). Os grafos (c), (d) e (e) são componentes de G .

Explicamos agora o comportamento do algoritmo em cada possível caso do subgrafo H .

Se o subgrafo H tem dois triângulos T' e T'' que formam uma borboleta com um triângulo que não está em H , então, como provaremos na seção 3.3.3, H é isomorfo a um dos grafos (a), (b) ou (c) da Figura 3.16. Em todos esses casos o algoritmo executa a redução apropriada. Devido à estrutura especial do grafo H , o algoritmo é capaz de decidir rapidamente se existe uma coleção $\mathcal{T} \subseteq \mathcal{T}_H$, $|\mathcal{T}| > 2$, localmente ótima em G . Se tal coleção existe, a redução adiciona \mathcal{T} a A^* e atualiza G . Caso contrário, o algoritmo reduz H de modo que no grafo reduzido os triângulos obtidos pela redução têm grau no máximo 3. O algoritmo decide qual destas reduções aplicar comparando as cardinalidades dos empacotamentos máximos de $\mathcal{T}_V(H)$ em casos quando ambos T' e T'' estão no empacotamento, quando exatamente um deles está no empacotamento, e quando nenhum deles está no empacotamento. Para dar uma intuição das reduções, apresentamos alguns exemplos nas Figuras 3.3 e 3.4.

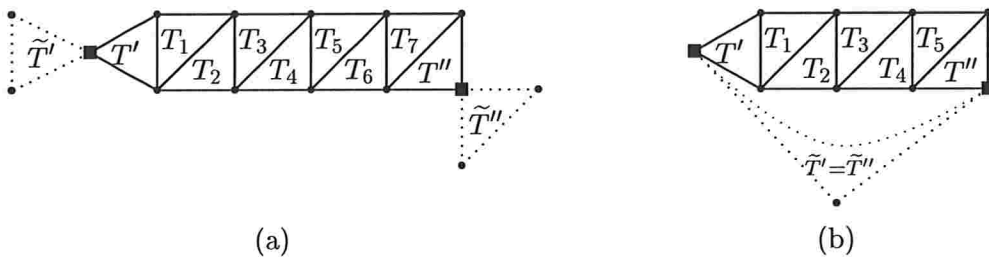


Figura 3.3: Exemplos do grafo H obtido no algoritmo $VT4_k$ (H é o grafo com arestas cheias). T' (resp. T'') é um triângulo de H que forma uma borboleta com um triângulo \tilde{T}' (resp. \tilde{T}'') que não está em H . Note que possivelmente $\tilde{T}' = \tilde{T}''$. (a) A coleção $\{T_1, T_4, T_7\}$ de $\mathcal{T}_V(H)$ é localmente ótima em G . (b) A coleção $\{T', T_3, T''\}$ de $\mathcal{T}_V(H)$ é localmente ótima em G .

Note que para o exemplo da Figura 3.3(a), os empacotamentos máximos de $\mathcal{T}_V(H)$ em todos os casos possíveis (quando ambos T', T'' estão no empacotamento, quando exatamente um deles está no empacotamento, e quando nenhum deles está no empacotamento) têm a mesma cardinalidade. Observe, ainda, que esses empacotamentos podem ser encontrados

facilmente. Por exemplo, um empacotamento máximo de $\mathcal{T}_V(H)$ no caso em que T' está no empacotamento e T'' não está no empacotamento pode ser obtido tomando-se o triângulo T' , o triângulo T_3 (que é localmente ótimo em $H - v_{T''\tilde{T}''} - V_{T'}$) e o triângulo T_6 (que é localmente ótimo em $H - v_{T''\tilde{T}''} - V_{T'} - V_{T_3}$). Provaremos que neste caso um empacotamento máximo de $\mathcal{T}_V(H)$ que não contém T' nem T'' é uma coleção localmente ótima em G (no exemplo, este empacotamento é $\{T_1, T_4, T_7\}$).

A Figura 3.3(b) mostra um caso para o qual $\tilde{T}' = \tilde{T}''$ e os empacotamentos máximos de $\mathcal{T}_V(H)$ não são de mesma cardinalidade nos três seguintes casos: quando ambos T' e T'' estão no empacotamento, quando exatamente um deles está no empacotamento, e quando nenhum deles está no empacotamento. Provaremos que neste caso um empacotamento máximo de $\mathcal{T}_V(H)$ que contém ambos T' e T'' é uma coleção localmente ótima em G (no exemplo, este empacotamento é $\{T', T_3, T''\}$).

Quanto ao exemplo da Figura 3.4, comparando os empacotamentos máximos de $\mathcal{T}_V(H)$ em todos os casos possíveis (quando ambos T' e T'' estão no empacotamento, quando exatamente um deles está no empacotamento, e quando nenhum deles está no empacotamento), notamos o seguinte. Mesmo que os empacotamentos em (c) e (e) tenham a mesma cardinalidade, a solução em (e) é “melhor solução” que a solução em (c) (pois se escolhermos $\{T', T_5\}$ para incluímos na solução final, precisamos omitir o triângulo \tilde{T}' , o que não é o caso se escolhermos $\{T_1, T_4\}$ para incluímos na solução final do algoritmo). Similarmente, mesmo que os empacotamentos em (d) e (e) sejam de mesmo tamanho, incluir $\{T_1, T_4\}$ é melhor que incluir $\{T_1, T''\}$ na solução final. Portanto, é sempre bom escolher $\{T', T_3, T''\}$ ou $\{T_1, T_4\}$ (e incluí-los na solução final do algoritmo). Entretanto, não sabemos qual dos dois é melhor globalmente. Portanto, aplicamos *Redução 1*(H), ou seja, substituímos todos os triângulos de H , exceto T' e T'' , por um novo triângulo T_H . Aplicamos a redução para eliminar os triângulos de grau maior que 3, isto é, para podermos aplicar o algoritmo $MIS3_k$. Como vamos ver na seção 3.3.3, essa redução preserva a razão de aproximação do algoritmo.

Se o subgrafo H tem somente um triângulo T' tal que existe um triângulo em $\mathcal{T}_G \setminus \mathcal{T}_H$ que forma uma borboleta com T' em G , então, como vamos provar na seção 3.3.3, H é isomorfo ao grafo da Figura 3.16(d). Além disso, $G[V_H]$ é uma componente de G . Observe que para o grafo $G[V_H]$ é fácil encontrar um empacotamento máximo: tome T', T_3, T_6, T_9 , etc. Note que T' é localmente ótimo em H , T_3 é localmente ótimo em $H - V_{T'}$, T_6 é localmente ótimo em $H - V_{T'} - V_{T_3}$, etc.

Se, no entanto, H não tem nenhum triângulo T' que forma uma borboleta com um triângulo em $\mathcal{T}_G \setminus \mathcal{T}_H$, então H é isomorfo ao grafo da Figura 3.16(e). Ademais, todo vértice tem grau 4, e portanto, $G[V_H]$ é uma componente de G . Note que se \tilde{T} é um triângulo em H , então $H - V_{\tilde{T}}$ é isomorfo ao grafo das arestas cheias mostrado na Figura 3.16(d). Mostraremos, portanto, que uma solução ótima do grafo H isomorfo ao grafo da Figura 3.16(e) pode ser obtido tomando-se qualquer triângulo \tilde{T} de H e uma solução ótima de $H - V_{\tilde{T}}$. Esta última pode ser construída da mesma forma como fizemos para o grafo da Figura 3.16(d).

Repetimos as iterações enquanto existe um triângulo em G com grau maior que 3. Quando não existe mais tal triângulo, aplicamos o algoritmo $MIS3_k$ no grafo de interseção de \mathcal{T}_G . Finalmente, para cada aplicação (em ordem reversa) de *Redução*(H), fazemos *Restauração*(H), ou seja, adicionamos os triângulos apropriados de H à solução final do algoritmo (veja o exemplo da Figura 3.4).

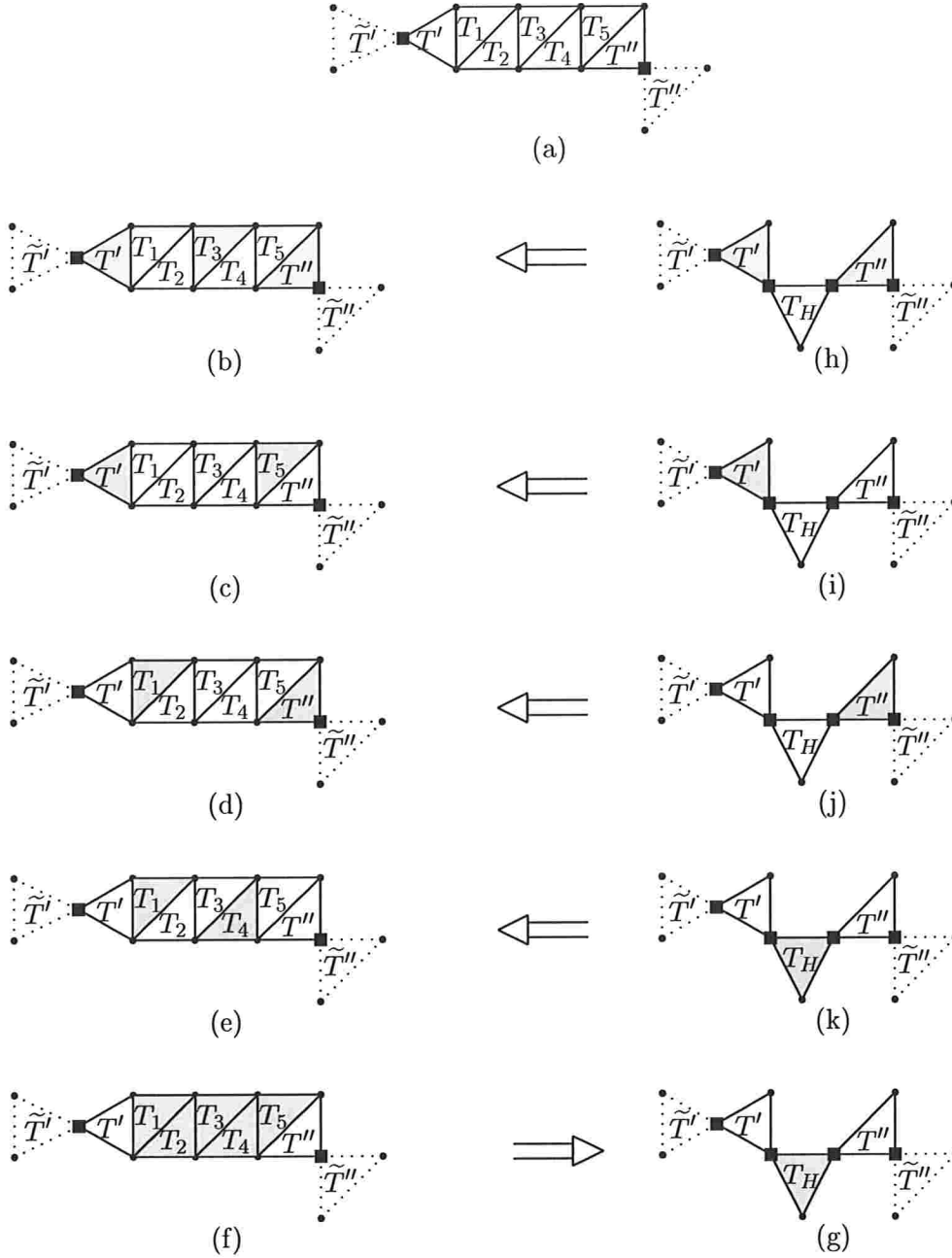


Figura 3.4: (a) Exemplo de um grafo H obtido no algoritmo $VT4_k$ (H é o grafo com arestas cheias). $T' \cup \tilde{T}'$ e $T'' \cup \tilde{T}''$ são borboletas em G . Dependendo se T', T'' estão ou não num empacotamento máximo de $\mathcal{T}_V(H)$, temos um dos casos (b), (c), (d) ou (e). (b) $\{T', T_3, T''\}$ é um empacotamento máximo de $\mathcal{T}_V(H)$ que contém T' e T'' . (c) $\{T', T_5\}$ é um empacotamento máximo de $\mathcal{T}_V(H)$ que contém T' mas não T'' . (d) $\{T_1, T''\}$ é um empacotamento máximo de $\mathcal{T}_V(H)$ que contém T'' mas não T' . (e) $\{T_1, T_4\}$ é um empacotamento máximo de $\mathcal{T}_V(H)$ que não contém T' nem T'' . (f) e (g) representam $Redução\ 1(H)$: ela substitui todos os triângulos de H , exceto T' e T'' , por um novo triângulo T_H . (h) Se T', T'' estão na solução do algoritmo antes da aplicação de $Restauração(H)$, então o procedimento $Restauração(H)$ adiciona T_3 à solução final. (i) Se T' está na solução do algoritmo antes da aplicação de $Restauração(H)$, mas T'' não, então este procedimento adiciona T_5 à solução final. (j) Semelhante ao caso (i). (k) Se T_H está na solução do algoritmo antes da aplicação de $Restauração(H)$, então o procedimento $Restauração(H)$ remove T_H da solução e adiciona T_1 e T_4 à solução.

Apresentamos agora um pseudocódigo do algoritmo $VT4_k$.

Algoritmo $VT4_k$

Entrada: Um grafo G irredundante de grau máximo 4.

```

1  $\mathcal{A}^* \leftarrow \emptyset$ 
2 ENQUANTO existe um triângulo em  $G$  com grau maior que 3
3   FAÇA ENQUANTO existe  $\mathcal{T} \subseteq \mathcal{T}_G$ ,  $|\mathcal{T}| \leq 2$ , localmente ótimo em  $G$ 
4     FAÇA  $Aceita(\mathcal{T})$ 
5     SE existe um triângulo  $T \in \mathcal{T}_G$  com  $d_G(T) > 3$ 
6       ENTÃO  $H \leftarrow$  subgrafo maximal conexo irredundante de  $G$  que
7         contém  $T$  e não contém nenhuma borboleta
8          $B_H \leftarrow \{T' \in \mathcal{T}_H : \text{existe um triângulo em } \mathcal{T}_G \setminus \mathcal{T}_H$ 
9           que forma uma borboleta com  $T'$  em  $G\}$ 
10        SE  $|B_H|$ 
11          é igual a 2:  $Redução(H)$ 
12          é igual a 1:  $Sol_H \leftarrow Adota(H)$ ,  $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup Sol_H$ 
13          é igual a 0: tome um triângulo  $\tilde{T}$  em  $\mathcal{T}_H$ ,
14             $Sol_H \leftarrow \{\tilde{T}\} \cup Adota(H - V_{\tilde{T}})$ ,  $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup Sol_H$ 
15 SE  $G \neq \emptyset$  ENTÃO  $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup MIS3_k(\text{grafo de interseção de } \mathcal{T}_G)$ 
16 PARA cada aplicação (em ordem reversa) de  $Redução(H)$  FAÇA  $Restauração(H)$ 
17 DEVOLVA  $\mathcal{A}^*$ 

```

Descrevemos em seguida cada procedimento do algoritmo em mais detalhes.

1. $Aceita(\mathcal{T})$: Adicione \mathcal{T} a \mathcal{A}^* , e remova os vértices de todos os triângulos em \mathcal{T} do grafo G . (O algoritmo executa $Aceita(\mathcal{T})$ não somente na linha 4, mas também em $Redução(H)$. Vamos ver mais adiante que o algoritmo executa $Aceita(\mathcal{T})$ somente se a coleção \mathcal{T} é localmente ótima.)
2. $Adota(H)$: Faça $\mathcal{E} := \emptyset$. Enquanto $H \neq \emptyset$, encontre um triângulo T localmente ótimo em H , adicione T a \mathcal{E} e remova V_T de H . Devolva \mathcal{E} . (Referimo-nos a $Adota(H)$ como sendo o conjunto \mathcal{E} devolvido por esse procedimento.)
3. $Redução(H)$: dado um grafo H como definido no algoritmo, com $|B_H| = 2$, este procedimento reduz H de modo que no grafo reduzido os triângulos obtidos pela redução têm grau no máximo 3 (no caso da aplicação de $Redução1(H)$ ou $Redução2(H)$), ou adiciona alguns triângulos de H a \mathcal{A}^* e atualiza o grafo (isto é, executa $Aceita(\mathcal{T})$, para um $\mathcal{T} \subseteq \mathcal{T}_H$). Mais formalmente, o procedimento $Redução(H)$ é o seguinte.

Sejam $T', T'' \in B_H$ e $\tilde{T}', \tilde{T}'' \in \mathcal{T}_G \setminus \mathcal{T}_H$ tais que $T' \cup \tilde{T}'$ e $T'' \cup \tilde{T}''$ são borboletas em G (possivelmente $\tilde{T}' = \tilde{T}''$). Considere os conjuntos:

$$\begin{aligned} Sol_{T'T''} &:= \{T', T''\} \cup Adota(H - V_{T'} - V_{T''}), & Sol_{T'\tilde{T}''} &:= \{T'\} \cup Adota(H - V_{T'} - v_{T''\tilde{T}''}), \\ Sol_{\tilde{T}'T''} &:= \{T''\} \cup Adota(H - V_{T''} - v_{T'\tilde{T}'}), & Sol_{\tilde{T}'\tilde{T}''} &:= Adota(H - v_{T'\tilde{T}'} - v_{T''\tilde{T}''}). \end{aligned}$$

Veremos na seção 3.3.3 que $Sol_{T'T''}$ é um empacotamento máximo de $\mathcal{T}_V(H)$ que contém T' e T'' , $Sol_{T'\tilde{T}''}$ é um empacotamento máximo de $\mathcal{T}_V(H)$ que contém T' mas não T'' , etc.

- (a) Se $|\text{Sol}_{T'T''}| = |\text{Sol}_{T'\tilde{T}''}| = |\text{Sol}_{\tilde{T}'T''}| = |\text{Sol}_{\tilde{T}'\tilde{T}''}|$ então aplique $\text{Aceita}(\text{Sol}_{\tilde{T}'\tilde{T}''})$. Veja a Figura 3.3(a).
- (b) Se a igualdade em (a) não está satisfeita e $\tilde{T}' = \tilde{T}''$ então aplique $\text{Aceita}(\text{Sol}_{T'T''})$. Veja a Figura 3.3(b).
- (c) Se $|\text{Sol}_{T'T''}| - 1 = |\text{Sol}_{T'\tilde{T}''}| = |\text{Sol}_{\tilde{T}'T''}| = |\text{Sol}_{\tilde{T}'\tilde{T}''}|$ e $\tilde{T}' \neq \tilde{T}''$ então aplique $\text{Redução 1}(H)$:

$$G \leftarrow (G - (E_H \setminus \{E_{T'} \cup E_{T''}\})) \cup T_H,$$

onde $T_H := [v', w, v'']$, w é um vértice novo, v' é qualquer vértice de T' diferente de $v_{T'\tilde{T}'}$, e v'' é qualquer vértice de T'' diferente de $v_{T''\tilde{T}''}$. Portanto, $\text{Redução 1}(H)$ substitui todos os triângulos de H , exceto T' e T'' , por um novo triângulo T_H que induz uma borboleta com T' e com T'' no grafo reduzido (veja as Figuras 3.4 (f) e (g)). Provaremos que T' e T'' são disjuntos, ou seja, $\text{Redução 1}(H)$ está bem definida.

- (d) Se $|\text{Sol}_{T'T''}| = |\text{Sol}_{T'\tilde{T}''}| = |\text{Sol}_{\tilde{T}'T''}| = |\text{Sol}_{\tilde{T}'\tilde{T}''}| + 1$ e $\tilde{T}' \neq \tilde{T}''$, então aplique $\text{Redução 2}(H)$:

$$G \leftarrow (G - E_H) \cup T_H^1 \cup T_H^2,$$

onde $T_H^1 := [v_{T'\tilde{T}'}, w_1, w]$, $T_H^2 := [w, w_2, v_{T''\tilde{T}''}]$ e w_1, w, w_2 são vértices novos. Logo, esta redução substitui todos os triângulos de H por dois triângulos novos T_H^1 e T_H^2 , tais que T_H^1 induz uma borboleta com T_H^2 e com \tilde{T}' ; e T_H^2 induz uma borboleta com \tilde{T}'' e com T_H^1 no grafo reduzido.

4. $\text{Restauração}(H)$: Se a redução aplicada a H foi $\text{Redução 1}(H)$ ou $\text{Redução 2}(H)$, então o procedimento $\text{Restauração}(H)$ adiciona os triângulos apropriados de H a \mathcal{A}^* .
- (a) Se a redução aplicada a H foi $\text{Redução 1}(H)$, então se T_H pertence a \mathcal{A}^* antes da aplicação de $\text{Restauração}(H)$, este procedimento remove T_H de \mathcal{A}^* e adiciona o conjunto $\text{Sol}_{\tilde{T}'\tilde{T}''}$ a ele (este conjunto foi computado no procedimento $\text{Redução}(H)$); se $T', T'' \in \mathcal{A}^*$, então $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup \text{Sol}_{T'T''}$; se $T' \in \mathcal{A}^*$, $T'' \notin \mathcal{A}^*$, então $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup \text{Sol}_{T'\tilde{T}''}$; e se $T' \notin \mathcal{A}^*$, $T'' \in \mathcal{A}^*$, então $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup \text{Sol}_{\tilde{T}'T''}$ (veja a Figura 3.4).
- (b) Se a redução aplicada a H foi $\text{Redução 2}(H)$, então se T_H^1 pertence a \mathcal{A}^* antes da aplicação de $\text{Restauração}(H)$, este procedimento adiciona $\text{Sol}_{T'\tilde{T}''}$ a \mathcal{A}^* e remove T_H^1 ; se $T_H^2 \in \mathcal{A}^*$, então adiciona $\text{Sol}_{\tilde{T}'T''}$ a \mathcal{A}^* e remove T_H^2 ; e se $T_H^1, T_H^2 \notin \mathcal{A}^*$, então adiciona $\text{Sol}_{\tilde{T}'\tilde{T}''}$ a \mathcal{A}^* .

3.3.3 Desempenho do algoritmo VT4_k

Observamos inicialmente o seguinte fato sobre o grafo H definido na linha 6 do algoritmo VT4_k.

Se H não contém borboletas, e T_1, T_2 são triângulos de H que têm somente um vértice z em comum, então existe pelo menos uma aresta de H com um extremo em $V_{T_1} \setminus \{z\}$ e outro em $V_{T_2} \setminus \{z\}$. (3.24)

Isso vale pois o grau máximo 4 já foi atingido no vértice z .

Lema 3.5 *Para cada iteração do algoritmo $VT4_k$ para a qual a condição da linha 5 está satisfeita, se $|B_H| = 2$, então os triângulos em B_H são disjuntos.*

Demonstração. Seja G o grafo na linha 5 do algoritmo $VT4_k$, H o subgrafo de G (como definido na linha 6) com $|B_H| = 2$, e $T', T'', \tilde{T}', \tilde{T}''$ como definidos no procedimento *Redução*(H).

Suponha que $|V_{T'} \cap V_{T''}| = 2$, e que $V_{T'} \cap V_{T''} = \{x, y\}$. Se $\tilde{T}' = \tilde{T}''$, então $T' \cup \tilde{T}'$ não é uma borboleta em G , uma contradição (veja a Figura 3.5(a)). Logo, $\tilde{T}' \neq \tilde{T}''$. Dos fatos que $v_{T'\tilde{T}'}$ e $v_{T''\tilde{T}''}$ já têm grau 4, $T' \cup \tilde{T}'$ (resp. $T'' \cup \tilde{T}''$) é uma borboleta em G e $\Delta(G) = 4$, concluímos que existe no máximo mais um triângulo em G que intersecta x ou y , a saber, o triângulo $[x, y, z]$. Note que agora x e y têm grau 4. Como H não contém borboletas, $[x, y, z]$ é o único triângulo de H que contém z (note que z pode ser um vértice na interseção de dois triângulos que formam uma borboleta). Mostramos portanto que H contém T', T'' e no máximo mais um triângulo $[x, y, z]$ (veja a Figura 3.5(b)). Mas isso implica que H não tem um triângulo T com $d_G(T) > 3$, uma contradição.

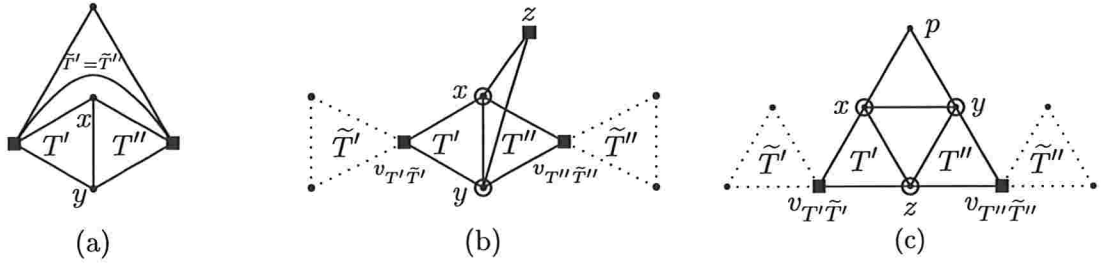


Figura 3.5: Os triângulos em B_H não se intersectam.

Suponha agora que T' e T'' têm somente um vértice em comum. Similarmente como acima, temos que $\tilde{T}' \neq \tilde{T}''$. Se $T' = \{v_{T'\tilde{T}'}, x, z\}$ e $T'' = \{v_{T''\tilde{T}''}, z, y\}$, como $v_{T'\tilde{T}'}$ e $v_{T''\tilde{T}''}$ já têm grau 4, usando o fato (3.24), temos que $xy \in E_H$. Agora, dos fatos que H tem um triângulo T com $d_G(T) > 3$, z tem grau 4 e $\Delta(G) = 4$, concluímos que existe no máximo um triângulo em G que intersecta x ou y , a saber, o triângulo $[x, y, p]$ (veja a Figura 3.5(c)). Mas agora, ambos x e y têm grau 4 e $[x, y, z]$ é localmente ótimo em G . Portanto, o algoritmo teria aplicado *Aceita*($[x, y, z]$) na linha 4, e novamente temos uma contradição. \square

Corolário 3.6 *O procedimento $Redução1(H)$ está bem definido e não cria novos triângulos em G exceto T_H .*

Demonstração. Segue diretamente do Lema 3.5. \square

Observamos ainda que *Redução2*(H) não cria novos triângulos em G , exceto T_H^1 e T_H^2 , pois $\tilde{T}' \neq \tilde{T}''$.

É fácil ver que após a execução de *Redução1*(H) ou *Redução2*(H), o grafo obtido ainda tem grau máximo 4 e é irredundante. Mostraremos que todas as outras reduções aplicadas no algoritmo consistem em adicionar a \mathcal{A}^* uma coleção \mathcal{T} localmente ótima em G e remover de G os vértices de todos os triângulos em \mathcal{T} . Portanto, as propriedades estruturais do grafo de entrada (grau máximo 4 e irredundância) são mantidas em cada iteração.

Além disso, observamos que

para cada iteração do algoritmo para a qual existe um triângulo em G com grau maior que 3, onde G é o grafo na linha 5, não existe nenhuma coleção \mathcal{T} , $|\mathcal{T}| \leq 2$, localmente ótima em G . (3.25)

Caso contrário, é claro que o algoritmo aplicaria $Aceita(\mathcal{T})$ na linha 4.

Lema 3.7 *Em cada iteração do algoritmo VT4_k, temos que $d_G(x) \geq 3$ para todo $x \in V_H$, onde H é o grafo na linha 6.*

Demonstração. Suponha que H tenha um vértice x com $d_G(x) = 2$. Seja T o triângulo em H que contém x . Se somente um triângulo de G fosse adjacente a T , então T seria localmente ótimo em G , uma contradição com (3.25). Portanto, $d_G(T) \geq 2$. Se existem dois triângulos T_1 e T_2 disjuntos nos vértices que intersectam T , então $T \cup T_1$ e $T \cup T_2$ são borboletas em G , ou seja, $H = T$, o que é impossível, pois H tem um triângulo com grau maior que 3 em G . Portanto, os triângulos que têm um vértice em comum com T se intersectam dois a dois, isto é, T é localmente ótimo em G , o que contradiz (3.25). \square

Teorema 3.8 *Em cada iteração do algoritmo VT4_k, temos que $|B_H| \leq 2$.*

Demonstração. Suponha que $B_H \neq \emptyset$ e considere T um triângulo em B_H . Pela definição de B_H , o triângulo T forma uma borboleta com um triângulo $\tilde{T} \notin \mathcal{T}_H$.

Se $d_H(T) = 4$, então usando os fatos de que $\Delta(G) = 4$ e $T \cup \tilde{T}$ é uma borboleta em G , concluímos que o subgrafo H induzido pelo triângulo T e pelos triângulos de H que intersectam T é isomorfo ao grafo da Figura 3.6(a). Note que y e z são saturados. Ademais, como $\Delta(G) = 4$, o vértice t (resp. w) do grafo mostrado na Figura 3.6(a) não está na interseção de dois triângulos que formam uma borboleta em G . Logo, se H é isomorfo ao grafo da Figura 3.6(a), então $[y, t, w]$ é localmente ótimo em G , o que contradiz (3.25). Portanto, como $v_{T\tilde{T}}$, y e z já são saturados, existe um vértice p adjacente a t e w . Note que agora t e w são saturados também. Do Lema 3.7 segue que $d_G(p) \geq 3$, ou seja, o triângulo $[t, p, w]$ também está em B_H , e portanto $|B_H| = 2$. Observe que H é isomorfo ao grafo da Figura 3.6(b).

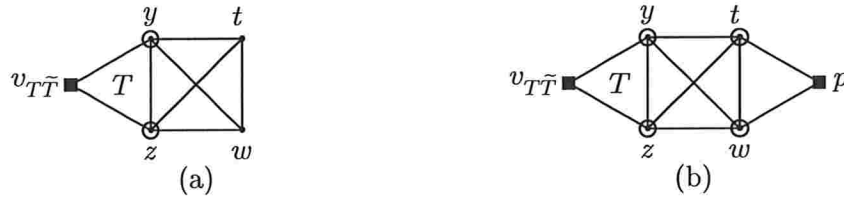


Figura 3.6: Os possíveis casos quando $d_H(T) = 4$.

Se, entretanto, $d_H(T) = 3$, então como $\Delta(G) = 4$ e $T \cup \tilde{T}$ é uma borboleta em G , o subgrafo de H induzido pelo triângulo T e pelos triângulos de H que intersectam T é isomorfo ao grafo da Figura 3.7(a) (note que y , p e z são saturados). Se H é isomorfo ao grafo da Figura 3.7(a), então $[y, p, z]$ é localmente ótimo em G , uma contradição. Portanto,

$tw \in E_H$. Se $d_G(t) = d_G(w) = 3$, então $[t, p, w]$ é localmente ótimo em G , novamente uma contradição. Logo, existe um vértice l adjacente a t e a w . Note que agora t e w são saturados também. Do Lema 3.7 segue que $d_G(l) \geq 3$, isto é, o triângulo $[t, l, w]$ também pertence a B_H , e portanto $|B_H| = 2$. Observe que H é isomorfo ao grafo da Figura 3.7(b).

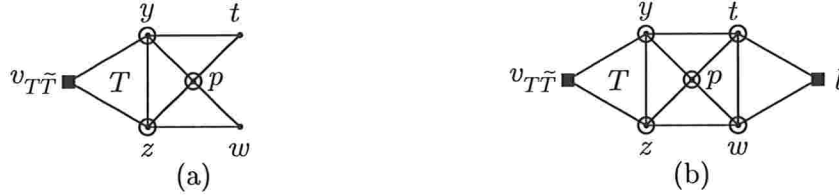


Figura 3.7: Os possíveis casos quando $d_H(T) = 3$.

Se $d_H(T) = 2$ e o subgrafo de H induzido pelo triângulo T e pelos triângulos de H que intersectam T é isomorfo ao grafo da Figura 3.8(a), então claramente $tw \notin E_H$. Além disso, $T \cup \tilde{T}$ é uma borboleta em G e $v_{T\tilde{T}}, z, y$ são saturados. Logo, do Lema 3.7 temos que $[y, t, z]$ (resp. $[y, w, z]$) forma uma borboleta com um triângulo que não está em H , com o vértice t (resp. w) na interseção dos triângulos que formam tal borboleta. Portanto, da definição de H , segue que neste caso H é isomorfo ao grafo da Figura 3.8(a). Mas então H não tem nenhum triângulo de grau maior que 3 em G , uma contradição. Logo, o subgrafo de H induzido por T e pelos triângulos de H que intersectam T é isomorfo ao grafo da Figura 3.8(b).



Figura 3.8: Se $d_H(T) = 2$, o subgrafo de H induzido por T e pelos triângulos de H que intersectam T não é isomorfo ao grafo em (a), mas ao grafo em (b). Em (b), o vértice y é saturado, pois $d_H(T) = 2$.

Agora, como $T \cup \tilde{T}$ é uma borboleta em G , $\Delta(G) = 4$, G é irredundante e H não contém borboletas, temos as seguintes duas possibilidades.

- H é isomorfo ao grafo da Figura 3.9(a). Note que H pode ter um número par ou ímpar de triângulos (a configuração do grafo H com número par de triângulos é similar à do grafo da Figura 3.9(a)). Como H tem um triângulo com grau em G maior que 3, o grafo H tem pelo menos 7 vértices. Ademais, pelo Lema 3.7, $d_G(x) \geq 3$, ou seja, $|B_H| = 2$ para esse grafo.
- H é isomorfo ao grafo da Figura 3.9(b) (linhas cheias indicam arestas em H). Note que H pode ter número par ou ímpar de triângulos (a configuração do grafo H com número par de triângulos é similar à do grafo da Figura 3.9(b)). Observe ainda que $xy \notin E_G$ e

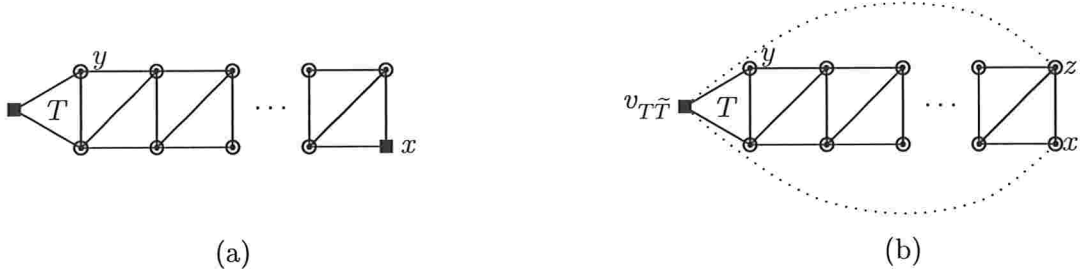


Figura 3.9: Configurações possíveis quando $d_H(T) = 2$. Em (a), $|B_H| = 2$ e o grafo H tem pelo menos 7 vértices. Em (b), as linhas cheias indicam arestas em H . O grafo (b) tem pelo menos 9 vértices, $G[V_H]$ é uma componente de G , xy não é aresta de G e $|B_H| = 1$.

que T forma uma borboleta com o triângulo $[v_{T\tilde{T}}, z, x]$. Mostramos em seguida que

$$G[V_H] \text{ é uma componente de } G. \quad (3.26)$$

Suponha que (3.26) não ocorra. Como todos os vértices do grafo da Figura 3.9(b) têm grau 4 em G , exceto x e y , então existe um vértice $w \notin V_H$ tal que, sem perda de generalidade, $xw \in E_G$. Da limitação do grau dos vértices do grafo e o fato de que $xy \notin E_G$, concluímos que xw não é aresta de nenhum triângulo, uma contradição. Logo, a afirmação (3.26) é válida. Temos então que $|V_H| \geq 9$ (caso contrário, H teria uma coleção \mathcal{T} , $|\mathcal{T}| \leq 2$, localmente ótima em G). Note ainda que $|B_H| = 1$.

Provamos portanto que $|B_H| \leq 2$ em todos os casos possíveis. \square

Lema 3.9 *Em cada iteração do algoritmo VT4_k, se $B_H = \emptyset$ então $d_H(x) \geq 3$ para cada $x \in V_H$ e H é uma componente de G .*

Demonstração. Suponha que exista $x \in V_H$ com $d_H(x) = 2$. Então o Lema 3.7 implica que existe uma aresta xw que não está em E_H . Do fato que G é irredundante, temos que xw é aresta de um triângulo $\tilde{T} \in \mathcal{T}_G \setminus \mathcal{T}_H$. Como $B_H = \emptyset$, então \tilde{T} não induz nenhuma borboleta com um triângulo de H . Além disso, x está em V_H e portanto, pela definição de H , temos que $\tilde{T} \in \mathcal{T}_H$, uma contradição. Similarmente, temos que H é uma componente de G . \square

Teorema 3.10 *Para cada iteração do algoritmo VT4_k, se $|B_H| \leq 1$, então $G[V_H]$ é uma componente de G e Sol_H é uma solução ótima nessa componente (onde Sol_H é o conjunto como definido nas linhas 12 e 13 do algoritmo VT4_k).*

Demonstração. Como vimos na prova do Teorema 3.8, se $|B_H| = 1$, então H é isomorfo ao grafo da Figura 3.9(b) e $G[V_H]$ é uma componente de G . Note que neste caso $\text{Adota}(H)$ é um empacotamento de $\mathcal{T}_V(H)$ que cobre todos os vértices de grafo H , exceto no máximo dois, e portanto $\text{Adota}(H)$ é uma solução ótima no componente $G[V_H]$.

Suponha agora que $B_H = \emptyset$. Do Lema 3.9, segue que H é uma componente de G , e portanto $|V_H| \geq 9$ (caso contrário, G teria uma coleção \mathcal{T} , $|\mathcal{T}| \leq 2$, localmente ótima).

Se cada par de triângulos em H tem dois vértices em comum, então H é isomorfo ao grafo da Figura 3.10. Portanto, H não tem nenhum triângulo T com $d_G(T) > 3$, uma contradição.

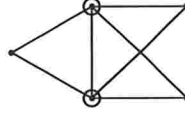


Figura 3.10: O caso em que todo par de triângulos em H tem dois vértices em comum não é possível.

Sejam então $T_1 = [x, y, z]$ e $T_2 = [x, t, r]$ triângulos em H com somente um vértice em comum. Como H não tem borboletas, usando o fato (3.24), podemos supor, sem perda de generalidade, que $yt \in E_H$. Agora, temos as seguintes possibilidades.

- (i) $rz \in E_H$. Como $|V_H| > 5$, usando a limitação do grau dos vértices do grafo e o fato de que o grafo é irredundante, temos que existe um vértice p adjacente a, sem perda de generalidade, y e a z . Do Lema 3.9, temos que $d_H(p) \geq 3$. Portanto, como H é irredundante e não tem borboletas, e x, y, z já são saturados, segue que $pt \in E_H$ ou $pr \in E_H$. Se ambos pt e pr são arestas de H , então todos os vértices de H são saturados, e $|V_H| \leq 8$, uma contradição com o fato de que $|V_H| \geq 9$. Se, no entanto, $pt \in E_H$, $pr \notin E_H$ (veja a Figura 3.11) então não existe um vértice k adjacente a r ou a p . De fato, suponha que exista um novo vértice k tal que, sem perda de generalidade, $rk \in E_G$. Da limitação do grau dos vértices e do fato de que $pr \notin E_H$, segue que rk não é aresta de nenhum triângulo, uma contradição. Portanto, novamente, $|V_H| \leq 8$, uma contradição. Similarmente se $pt \notin E_H$, $pr \in E_H$.

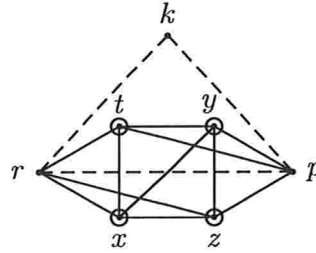


Figura 3.11: O grafo H quando rz, pt estão em E_H , e pr não está em E_H . As linhas tracejadas não podem ser arestas de H .

- (ii) $rz \notin E_H$, $ry \in E_H$. Se $tz \in E_H$, então claramente x, y, t são saturados. Como $rz \notin E_H$, similarmente como acima, concluímos que não existe vértice k adjacente a r ou a z . Portanto, H é isomorfo ao grafo da Figura 3.12(a) e $|V_H| \leq 8$, uma contradição. Se, entretanto, $tz \notin E_H$, então de (3.25) temos que $|V_H| > 5$. Então, como $tz, rz \notin E_H$, o único jeito de expandir o grafo é por um novo vértice w adjacente a t e a r . Agora, do Lema 3.9, temos que $d_H(z) \geq 3$ e $d_H(w) \geq 3$. Como x, y, r, t já são saturados, a única possível configuração de H é mostrada na Figura 3.12(b). Mas então H tem uma borboleta $[r, t, w] \cup [w, k, z]$, uma contradição.

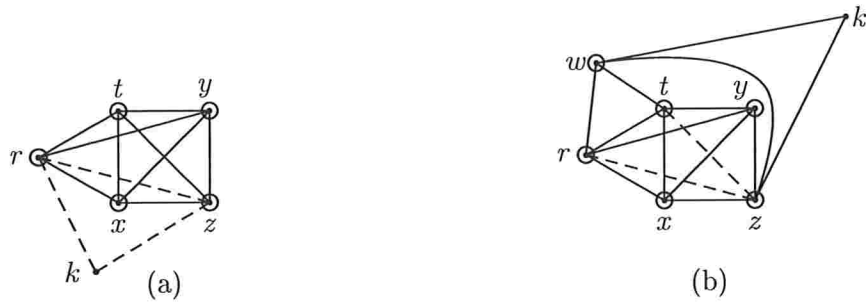


Figura 3.12: Os casos em que rz não está em E_H , e ry está em E_H .

- (iii) $rz, ry \notin E_H, tz \in E_H$. Note que este caso é equivalente ao caso (ii) quando $tz \notin E_H$.
- (iv) $rz, ry, tz \notin E_H$.

(1) Existe um vértice p adjacente a t e a y em H . Como x, y, t são saturados, do Lema 3.9 e do fato de que H não tem borboletas, temos que $rp, pz \in E_H$, e então novamente $|V_H| \leq 8$, uma contradição (veja a Figura 3.13).

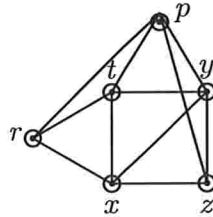


Figura 3.13: O caso em que rz, ry, tz não são arestas de H e existe um vértice p adjacente a t e a y em H .

(2) Não existe vértice adjacente a t e a y em H . Como $|V_H| > 5$, existe um vértice p adjacente a, sem perda de generalidade, y e a z . Como o caso (i) não é possível, temos que $tp \notin E_H$. De fato, se $tp \in E_H$, teríamos um grafo (o grafo da Figura 3.14(a) induzido pelos vértices x, y, z, t, p) isomorfo ao grafo da Figura 3.11 induzido pelos vértices x, y, z, t, r (o que é, como provamos, impossível).

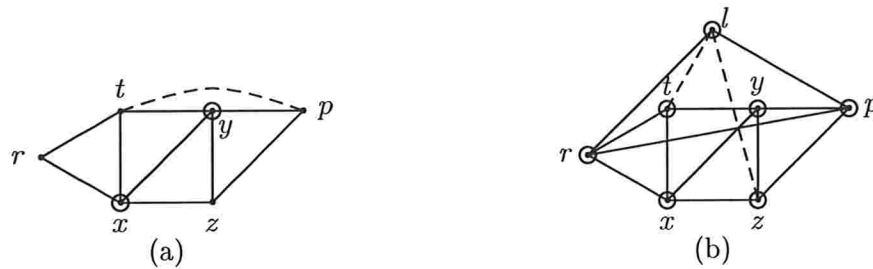


Figura 3.14: Casos em que rz, ry, tz não são arestas de H e não existe vértice adjacente a t e a y em H .

Suponha agora que não exista nenhum vértice adjacente a ambos p e z , ou adjacente a ambos r e t . Então, do Lema 3.9 e do fato de que $tp, rz \notin E_H$, segue que $pr \in E_H$.

Como H é irredundante e $tp, rz \notin E_H$, concluímos que existe um triângulo $[p, r, l]$ em H que contém a aresta pr . Agora, nossa hipótese (de que não existe vértice adjacente a ambos p e z , ou adjacente a ambos r e t) implica que $lt, lz \notin E_H$, e portanto o grafo H tem uma borboleta $[r, p, l] \cup [r, t, x]$ (veja a Figura 3.14(b)), uma contradição. Concluímos portanto que existe um vértice l adjacente a, sem perda de generalidade, p e a z (Figura 3.15(a)).

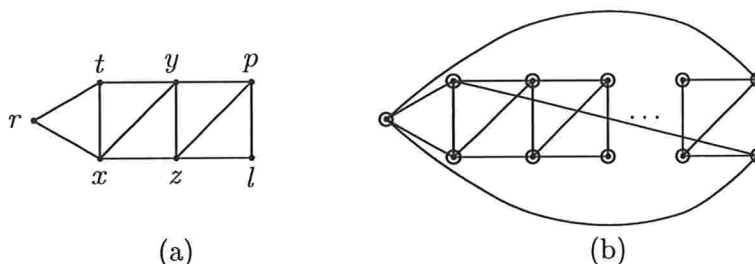


Figura 3.15: Um grafo H que tem o grafo em (a) como subgrafo e satisfaz $B_H = \emptyset$ é isomorfo ao grafo em (b).

Note que o grafo H que tem o grafo na Figura 3.15(a) como subgrafo e satisfaz $B_H = \emptyset$ é isomorfo ao grafo da Figura 3.15(b). Observe que H pode ter um número par ou ímpar de triângulos (a configuração do grafo H com número par de triângulos é similar à grafo da Figura 3.15(b)). Como todos os vértices nesse grafo são saturados, H é uma componente de G , e portanto $|V_H| \geq 9$ (caso contrário, H teria uma coleção \mathcal{T} , $|\mathcal{T}| \leq 2$, localmente ótima em G). Observe que, se \tilde{T} é um triângulo em H , então $H - V_{\tilde{T}}$ é isomorfo ao grafo com linhas cheias mostrado na Figura 3.9(b) (mas agora com pelo menos 6 vértices). Portanto, $Adota(H - V_{\tilde{T}})$ é um empacotamento de triângulos disjuntos nos vértices em $H - V_{\tilde{T}}$ que cobre todos os vértices de $H - V_{\tilde{T}}$, exceto no máximo dois. Como $|V_H| = |V_H - V_{\tilde{T}}| + 3$, temos que $\tilde{T} \cup Adota(H - V_{\tilde{T}})$ é uma solução ótima em H . \square

Corolário 3.11 *Em cada iteração do algoritmo $VT4_k$, o grafo H é isomorfo a um dos grafos na Figura 3.16.*

Demonstração. Segue das provas dos Teoremas 3.8 e 3.10. \square

Teorema 3.12 *Se \mathcal{T} é uma coleção localmente ótima em G , então $Aceita(\mathcal{T})$ preserva a razão de aproximação do algoritmo $VT4_k$.*

Demonstração. Notamos, inicialmente, que existe um empacotamento máximo de $\mathcal{T}_V(G)$ que contém \mathcal{T} . De fato, se um empacotamento máximo de $\mathcal{T}_V(G)$ não contém \mathcal{T} , podemos substituir os triângulos que intersectam algum triângulo em \mathcal{T} , pelos triângulos de \mathcal{T} . Logo, para $G' := G - \bigcup_{T \in \mathcal{T}} V_T$ temos $opt(G') \geq opt(G) - |\mathcal{T}|$. Portanto, se um empacotamento \mathcal{A} de $\mathcal{T}_V(G')$ satisfaz $opt(G') \leq \rho|\mathcal{A}|$ (com $\rho \geq 1$), então $opt(G) \leq opt(G') + |\mathcal{T}| \leq \rho|\mathcal{A}| + |\mathcal{T}| \leq \rho|\mathcal{A} \cup \mathcal{T}|$. \square

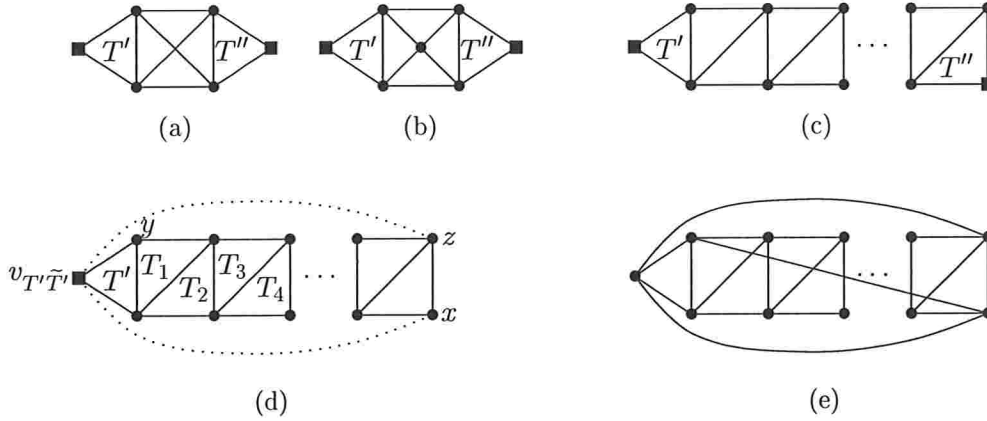


Figura 3.16: Possíveis configurações do grafo H . Cada vértice quadrado é comum a dois triângulos de G cuja união é uma borboleta. O grafo (c) tem pelo menos 7 vértices. Os grafos (d) e (e) têm pelo menos 9 vértices e $G[V_H]$ é uma componente de G (em (d) xy não é aresta de G). As linhas pontilhadas indicam arestas que estão em E_G , mas não em E_H . Note que, em (d), T' induz uma borboleta com o triângulo $[v_{T'\tilde{T}'}, x, z]$. Em (c), (d) e (e), H pode ter um número par ou ímpar de triângulos (a configuração do grafo H com número par de triângulos é similar ao grafo mostrado nesta figura).

Corolário 3.13 *Se \mathcal{T} é uma coleção localmente ótima em um grafo G e \mathcal{A} é um empacotamento máximo de triângulos disjuntos nos vértices em $G - \bigcup_{T \in \mathcal{T}} V_T$, então $\mathcal{A} \cup \mathcal{T}$ é um empacotamento máximo de $\mathcal{T}_V(G)$.*

Demonstração. Segue diretamente da prova do Teorema 3.12 (tome $\rho = 1$). \square

Teorema 3.14 *Redução(H) preserva a razão de aproximação do algoritmo $VT4_k$.*

Demonstração. Considere qualquer iteração do algoritmo para a qual existe um triângulo em G com grau maior que 3 (onde G é o grafo na linha 5) e $|B_H| = 2$. Então H é um dos grafos da Figura 3.16(a), (b) ou (c). Note que

$$\begin{aligned} &\text{se } K \text{ é qualquer um dos grafos } H - V_{T'} - V_{T''}, H - V_{T'} - v_{T''\tilde{T}''}, H - V_{T''} - v_{T'\tilde{T}'}, \\ &H - v_{T'\tilde{T}'} - v_{T''\tilde{T}''}, \text{ então } Adota(K) \text{ é um empacotamento máximo de } \mathcal{T}_V(K). \end{aligned} \quad (3.27)$$

Conseqüentemente, $Sol_{T'T''}$ é um empacotamento máximo de $\mathcal{T}_V(H)$ que contém T' e T'' , $Sol_{T'\tilde{T}''}$ é um empacotamento máximo de $\mathcal{T}_V(H)$ que contém T' mas não T'' , etc. Ademais, para H isomorfo a algum dos grafos da Figura 3.16(a), (b) ou (c), as igualdades listadas na definição da *Redução(H)* cobrem todas as possíveis relações entre as cardinalidades dos conjuntos $Sol_{T'T''}$, $Sol_{T'\tilde{T}''}$, $Sol_{\tilde{T}'T''}$ e $Sol_{\tilde{T}'\tilde{T}''}$.

Suponha que a igualdade em 3(a) seja satisfeita. Da definição do grafo H e do fato de que $T' \cup \tilde{T}'$ (resp. $T'' \cup \tilde{T}''$) é uma borboleta em G , concluímos que os triângulos em H que têm um vértice em comum com T' (resp. T'') se intersectam dois a dois. Além disso, vimos que T' e T'' são disjuntos, e conseqüentemente, $\{T', T''\}$ é localmente ótimo em H . Logo, da afirmação (3.27) (tome $K = H - V_{T'} - V_{T''}$) e do Corolário 3.13, segue que $Sol_{T'T''}$ é um empacotamento máximo de $\mathcal{T}_V(H)$. Como $|Sol_{T'T''}| = |Sol_{\tilde{T}'\tilde{T}''}|$, então $Sol_{\tilde{T}'\tilde{T}''}$ é também um empacotamento máximo de $\mathcal{T}_V(H)$. Note que $\{T \in \mathcal{T}_G : T \text{ intersecta um triângulo em } Sol_{\tilde{T}'\tilde{T}''}\} \subseteq$

\mathcal{T}_H . Além disso, $\mathcal{T}_H \subseteq \{T \in \mathcal{T}_G : T \text{ intersecta um triângulo em } \text{Sol}_{\tilde{T}'\tilde{T}''}\}$, pois, caso contrário, $\text{Sol}_{\tilde{T}'\tilde{T}''}$ não seria um empacotamento máximo de $\mathcal{T}_V(H)$. Segue que $\text{Sol}_{\tilde{T}'\tilde{T}''}$ é localmente ótimo em G , e pelo Teorema 3.12, $\text{Aceita}(\text{Sol}_{\tilde{T}'\tilde{T}''})$ preserva a razão de aproximação do algoritmo.

Suponha agora que a igualdade em 3(a) não é satisfeita e que $\tilde{T}' = \tilde{T}''$. Seja \mathcal{A} um empacotamento máximo de $\mathcal{T}_V(H \cup \tilde{T}')$. Se $\tilde{T}' \in \mathcal{A}$, então claramente, $|\mathcal{A}| = |\text{Sol}_{\tilde{T}'\tilde{T}''}| + 1$; se $\tilde{T}' \notin \mathcal{A}$, então $|\mathcal{A}| = |\text{Sol}_{T'T''}|$ (pois $|\text{Sol}_{T'T''}|, |\text{Sol}_{\tilde{T}'T''}|, |\text{Sol}_{T'\tilde{T}''}| \leq |\text{Sol}_{T'T''}|$). A igualdade em 3(a) não é satisfeita, ou seja, uma das igualdades em 3(c) ou 3(d) é satisfeita. Logo, $|\text{Sol}_{T'T''}| = |\text{Sol}_{\tilde{T}'\tilde{T}''}| + 1$. Portanto, $\text{Sol}_{T'T''}$ é um empacotamento máximo de $\mathcal{T}_V(H \cup \tilde{T}')$. Agora, analogamente como acima, concluímos que $\{T \in \mathcal{T}_G : T \text{ intersecta um triângulo em } \text{Sol}_{T'T''}\} = \mathcal{T}_H \cup \tilde{T}'$. Logo, $\text{Sol}_{T'T''}$ é localmente ótimo em G , e o Teorema 3.12 implica que $\text{Aceita}(\text{Sol}_{T'T''})$ preserva a razão de aproximação do algoritmo.

Se a igualdade em 3(c) ou 3(d) é satisfeita e $\tilde{T}' \neq \tilde{T}''$, definimos G_A como o grafo G antes da execução de *Redução 1*(H) (resp. *Redução 2*(H)), e G_P como o grafo G logo após a execução da *Redução 1*(H) (resp. *Redução 2*(H)). Seja ainda \mathcal{A}_P um empacotamento maximal de triângulos disjuntos nos vértices em G_P .

Suponha que a igualdade em 3(c) seja satisfeita e que $\tilde{T}' \neq \tilde{T}''$. Mostramos inicialmente que

$$\text{opt}(G_P) \geq \text{opt}(G_A) - |\text{Sol}_{\tilde{T}'\tilde{T}''}| + 1. \quad (3.28)$$

Seja \mathcal{T}_A^* um conjunto de triângulos que é um empacotamento máximo de $\mathcal{T}_V(G_A)$. Se $T', T'' \in \mathcal{T}_A^*$, então $|\mathcal{T}_A^* \cap \mathcal{T}_H| = |\text{Sol}_{T'T''}|$, pois $\text{Sol}_{T'T''}$ é um empacotamento máximo de $\mathcal{T}_V(H)$ que contém T' e T'' ; logo, $\mathcal{T}_A^* \setminus (\mathcal{T}_H \setminus \{T', T''\})$ é um empacotamento de $\mathcal{T}_V(G_P)$ cujo tamanho é $\text{opt}(G_A) - |\text{Sol}_{T'T''}| + 2 = \text{opt}(G_A) - |\text{Sol}_{\tilde{T}'\tilde{T}''}| + 1$. Se $T', T'' \notin \mathcal{T}_A^*$, então $|\mathcal{T}_A^* \cap \mathcal{T}_H| = |\text{Sol}_{\tilde{T}'\tilde{T}''}|$, e portanto $(\mathcal{T}_A^* \setminus \mathcal{T}_H) \cup \mathcal{T}_H$ é um empacotamento de $\mathcal{T}_V(G_P)$ de cardinalidade $\text{opt}(G_A) - |\text{Sol}_{\tilde{T}'\tilde{T}''}| + 1$. Se, entretanto, $T' \in \mathcal{T}_A^*$, $T'' \notin \mathcal{T}_A^*$, então $\mathcal{T}_A^* \setminus (\mathcal{T}_H \setminus T')$ é um empacotamento de $\mathcal{T}_V(G_P)$ de tamanho $\text{opt}(G_A) - |\text{Sol}_{T'\tilde{T}''}| + 1 = \text{opt}(G_A) - |\text{Sol}_{\tilde{T}'\tilde{T}''}| + 1$. Similarmente se $T' \notin \mathcal{T}_A^*$, $T'' \in \mathcal{T}_A^*$. Portanto, a afirmação (3.28) é válida.

Note que, como \mathcal{A}_P é maximal, pelo menos um dos triângulos T', T'', T_H está em \mathcal{A}_P . Definimos agora o conjunto \mathcal{A}_A (de acordo com *Restauração*(H)).

$$\mathcal{A}_A := \begin{cases} \mathcal{A}_P \setminus T_H \cup \text{Sol}_{\tilde{T}'\tilde{T}''} & \text{se } T_H \in \mathcal{A}_P \\ \mathcal{A}_P \cup \text{Sol}_{T'T''} & \text{se } T', T'' \in \mathcal{A}_P \\ \mathcal{A}_P \cup \text{Sol}_{T'\tilde{T}''} & \text{se } T' \in \mathcal{A}_P, T'' \notin \mathcal{A}_P \\ \mathcal{A}_P \cup \text{Sol}_{\tilde{T}'T''} & \text{se } T' \notin \mathcal{A}_P, T'' \in \mathcal{A}_P \end{cases}$$

Mostramos em seguida que

$$|\mathcal{A}_A| = |\mathcal{A}_P| + |\text{Sol}_{\tilde{T}'\tilde{T}''}| - 1 \text{ e} \quad (3.29)$$

$$\mathcal{A}_A \text{ é um empacotamento maximal em } \mathcal{T}_V(G_A). \quad (3.30)$$

Se $T_H \in \mathcal{A}_P$, então é claro que $|\mathcal{A}_A| = |\mathcal{A}_P| + |\text{Sol}_{\tilde{T}'\tilde{T}''}| - 1$. Se $T', T'' \in \mathcal{A}_P$, então $|\mathcal{A}_A| = |\mathcal{A}_P| + |\text{Sol}_{T'T''}| - 2 = |\mathcal{A}_P| + |\text{Sol}_{\tilde{T}'\tilde{T}''}| - 1$. Se, no entanto, $T' \in \mathcal{A}_P$, $T'' \notin \mathcal{A}_P$, então $|\mathcal{A}_A| = |\mathcal{A}_P| + |\text{Sol}_{T'\tilde{T}''}| - 1 = |\mathcal{A}_P| + |\text{Sol}_{\tilde{T}'\tilde{T}''}| - 1$. Similarmente se $T' \notin \mathcal{A}_P$, $T'' \in \mathcal{A}_P$. Logo, a afirmação (3.29) é válida. Da afirmação (3.27) e da maximalidade de \mathcal{A}_P , segue que \mathcal{A}_A é um empacotamento maximal de $\mathcal{T}_V(G_A)$ em todos os casos. Portanto, podemos concluir que (3.30) é válido.

Se $\text{opt}(G_P) \leq \rho|\mathcal{A}_P|$ (para algum $\rho \geq 1$), então de (3.28) e (3.29) temos $\text{opt}(G_A) \leq \text{opt}(G_P) + |\text{Sol}_{\overline{T'}\overline{T''}}| - 1 \leq \rho|\mathcal{A}_P| + |\text{Sol}_{\overline{T'}\overline{T''}}| - 1 = \rho(|\mathcal{A}_A| - |\text{Sol}_{\overline{T'}\overline{T''}}| + 1) + |\text{Sol}_{\overline{T'}\overline{T''}}| - 1$. Note que $|\text{Sol}_{\overline{T'}\overline{T''}}| \geq 1$ e portanto $\text{opt}(G_A) \leq \rho|\mathcal{A}_A|$.

Observamos que o algoritmo MIS3_k devolve um conjunto estável maximal e, portanto, imediatamente após a linha 15, \mathcal{A}^* é um empacotamento maximal de $\mathcal{T}_V(G)$. Notamos ainda que as restaurações dos subgrafos são feitas na ordem reversa das respectivas reduções. Portanto, usando (3.30), por indução podemos deduzir que, para cada *Redução 1(H)* aplicada no algoritmo, se G_P é o grafo como definido previamente, então \mathcal{A}_P é um empacotamento maximal de $\mathcal{T}_V(G_P)$. Segue, portanto, que *Restauração(H)* está bem definida, ou seja, pelo menos um dos triângulos T' , T'' , T_H pertence a \mathcal{A}^* antes da execução de *Restauração(H)*. Ademais, como vimos, \mathcal{A}_A é um empacotamento de $\mathcal{T}_V(G_A)$, e $\text{opt}(G_P) \leq \rho|\mathcal{A}_P|$ implica em $\text{opt}(G_A) \leq \rho|\mathcal{A}_A|$. Concluimos portanto que cada *Redução 1(H)* (e *Restauração(H)* correspondente) preserva a razão de aproximação do algoritmo.

A prova do fato de que *Redução 2(H)* (e *Restauração(H)* correspondente) preserva a razão de aproximação do algoritmo é análoga. \square

Teorema 3.15 *O algoritmo VT4_k é um algoritmo de $(3 - \frac{\sqrt{13}}{2} + \frac{13-\sqrt{13}}{52k})$ -aproximação para o problema VTP-4.*

Demonstração. De acordo com os Teoremas 3.10, 3.12 e 3.14, todas as reduções aplicadas no algoritmo VT4_k preservam a razão de aproximação. Portanto, a razão de aproximação do algoritmo VT4_k é a razão de aproximação do algoritmo MIS3_k. \square

3.3.4 Complexidade computacional do algoritmo VT4_k

Deve-se notar que o algoritmo VT4_k não precisa fazer uma busca exaustiva dos grafos mostrados na Figura 3.16. Se nenhuma chamada for feita a MIS3_k, o algoritmo VT4_k encontra uma solução ótima e pode ser implementado de modo a não consumir tempo maior que $O(n_G^3)$, onde n_G é a ordem do grafo de entrada. De fato, observe que para encontrar todos os triângulos em G gastamos tempo $O(n_G)$, pois basta, para cada vértice v de G , verificar se cada par de arestas que incide em v forma um triângulo. Como $d_G(v) \leq 4$, para cada vértice v temos no máximo 6 verificações. Note ainda que como $\Delta(G) \leq 4$, todo triângulo é adjacente a no máximo 9 triângulos. A linha 2 consome tempo $O(|\mathcal{T}_G|)$, pois, no começo do algoritmo, podemos colocar todos os triângulos com grau maior que 3 em G numa lista, gastando tempo $O(|\mathcal{T}_G|)$. Cada vez que o procedimento *Aceita* é executado ou alguma redução é feita no algoritmo, atualizamos esta lista em tempo constante. As linhas 3 e 4 consomem tempo $O(|\mathcal{T}_G|^3)$. De fato, a busca exaustiva de uma coleção $\mathcal{T} \subseteq \mathcal{T}_G$ com $|\mathcal{T}| \leq 2$ localmente ótimo em G consome tempo $O(|\mathcal{T}_G|^2)$; ainda, a linha 4 é executada no máximo $O(|\mathcal{T}_G|)$ vezes. Observe que o grafo H pode ser encontrado em tempo linear em $|\mathcal{T}_G|$. O procedimento *Adota(H)* consome tempo $O(|\mathcal{T}_G|)$. Isto porque encontrar um primeiro triângulo T localmente ótimo em H usando busca exaustiva consome tempo $O(|\mathcal{T}_G|)$. Devido à estrutura especial do grafo H , segue imediatamente que os candidatos para o próximo triângulo localmente ótimo em H são aqueles que são adjacentes a algum triângulo removido anteriormente. Finalmente, como

$\Delta(G) = 4$, do Lema 3.1 temos que $|\mathcal{T}_G| \leq 2n_G$. Logo, o consumo de tempo do algoritmo $VT4_k$ é $O(n_G^3)$, se nenhuma chamada for feita a $MIS3_k$.

O tempo de execução do algoritmo $VT4_k$ é dominado pela complexidade da rotina $MIS3_k$, que é $O(n^{O(k)})$. Portanto, qualquer melhoria na razão de aproximação e/ou tempo de execução do algoritmo para encontrar um subconjunto de vértices independentes de cardinalidade máxima em grafos de grau máximo 3 conduzirá a uma melhoria na qualidade do algoritmo $VT4_k$.

3.3.5 Observações finais

Mostramos em seguida que o algoritmo $VT4_k$ é superior ao algoritmo simples de busca local, apresentando instâncias do problema VTP para as quais a razão do valor de uma solução ótima sobre o valor devolvido pelo algoritmo de busca local de Hurkens e Schrijver é $\frac{3}{2} + \epsilon$.

Construiremos um grafo G cujos vértices serão os triângulos de um outro grafo G' (G' é a instância desejada para a qual o algoritmo de busca local atinge a razão $\frac{3}{2} + \epsilon$). A construção do grafo G que apresentaremos a seguir foi obtida por Yu e Goldschmidt [30], para mostrar que a razão de aproximação do algoritmo de busca local para o problema do conjunto independente máximo de vértices em grafos livre de k -garras é justa.

Seja r um inteiro fixo positivo. Começamos com conjuntos de vértices S_1 e T_1 com $|S_1| = |T_1| = 1$. Adicionamos conjuntos de vértices S_i , para $i = 2, \dots, r$, com $|S_i| = 2^{i-1}$, conjuntos de vértices T_2^i , para $i = 1, \dots, r-1$, com $|T_2^i| = |S_{i+1}|$, e finalmente um conjunto de vértices T_2^r com $|T_2^r| = |S_r|$. Todos os conjuntos de vértices são disjuntos. As arestas do grafo G são como segue. Existe uma aresta entre o único vértice de T_1 e o único vértice de S_1 , e uma correspondência unívoca entre os vértices de T_2^i e os vértices de S_{i+1} , para $i = 1, \dots, r-1$ (isto é, existe uma aresta entre cada par correspondente de vértices). Para $i = 1, \dots, r$, colocamos $2|S_i|$ arestas entre o conjunto de vértices S_i e o conjunto de vértices T_2^i , de modo que cada vértice no conjunto S_i tenha grau exatamente 3 e cada vértice em T_2^i tenha grau exatamente 2 (veja a Figura 3.17).

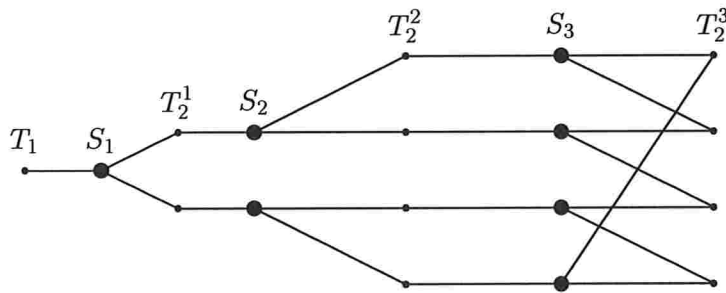
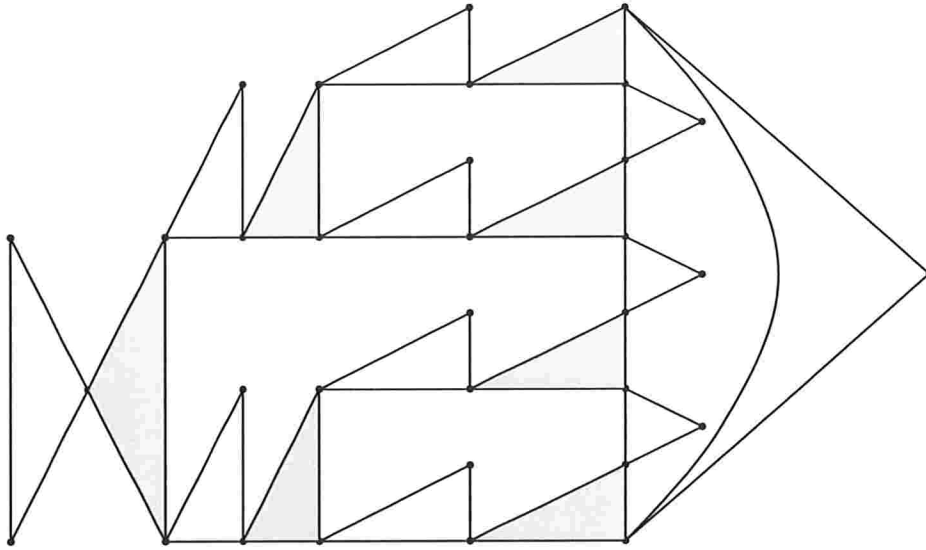


Figura 3.17: Exemplo do grafo G com $r = 3$.

Este grafo G é o grafo de interseção de $\mathcal{T}_{G'}$ (veja a Figura 3.18).

Figura 3.18: Exemplo de um grafo G' com $r = 3$.

Note que o conjunto de triângulos de G' que corresponde ao conjunto de vértices $I = \bigcup_{i=1}^r S_i$ de G é $(r + 1)$ -ótimo em G' . Isto é, para cada $p \leq r + 1$, a união de quaisquer p triângulos disjuntos nos vértices (dois a dois) em G' intersecta pelo menos p triângulos que correspondem ao conjunto I de G . No entanto, um empacotamento máximo de $\mathcal{T}_V(G')$ é o conjunto de triângulos que corresponde ao conjunto de vértices $I^* = T_1 \cup (\bigcup_{i=1}^r T_2^i)$ de G . Observe que a razão $|I^*|/|I|$ é $1 + \frac{2^{r-1}}{2^r-1} = \frac{3 \cdot 2^r - 2}{2 \cdot 2^r - 2}$. Portanto, a razão $|I^*|/|I|$ se aproxima de $3/2$ à medida que r aumenta.

Observamos ainda que o exemplo justo para o problema do empacotamento máximo de k -conjuntos apresentado por Hurkens e Schrijver [23] dá origem a uma instância do VTP, isto é, um grafo G , para o qual a razão do valor de $\text{opt}_{\text{VTP}}(G)$ sobre o valor da solução obtida pelo algoritmo $\text{HS}(\mathcal{T}_V(G), t)$ se aproxima de $3/2$ (quando t cresce), e G é irredundante com $\Delta(G) = 4$. No entanto, tal grafo G tem muito mais triângulos do que o exemplo de Yu e Goldschmidt [30] que apresentamos acima. Por exemplo, para $t = 4$ esse grafo tem 285 vértices e 152 triângulos (95 numa solução ótima e 57 na solução obtida pelo algoritmo de busca local).

3.4 Empacotamento de triângulos disjuntos nos vértices em grafos de intervalos unitários

3.4.1 Introdução

Um *grafo de intervalos* é um grafo de interseção de um conjunto finito de intervalos em \mathbb{R} . Quando todos esses intervalos têm o mesmo tamanho, temos um *grafo de intervalos unitários*. Quando nenhum dos intervalos contém qualquer outro, temos um *grafo de intervalos próprios*. Um grafo G é um *grafo de indiferença* (*indifference graph*) se existe um número positivo δ e uma função $f : V_G \rightarrow \mathbb{R}$, tal que, para todo par $u, v \in V_G$, $u \neq v$, uv é aresta de G se e somente se $|f(u) - f(v)| < \delta$.

Roberts [28] mostrou que as classes de grafos de indiferença, grafos de intervalos unitários e grafos de intervalos próprios são equivalentes.

Descrevemos a seguir um algoritmo linear para o problema VTP restrito a grafos de intervalos unitários. Esse algoritmo é baseado na seguinte caracterização, devida a Looges e Olariu [26]: um grafo G é um grafo de intervalos unitários se, e somente se, existe uma ordenação linear $<$ de vértices de G tal que, para cada escolha de vértices u, v, w em V_G temos que

$$\text{se } u < v < w \text{ e } uv \in E_G, \text{ então } vw \in E_G. \quad (3.31)$$

Essa ordenação $<$ será chamada *canônica*.

3.4.2 Algoritmo exato

Apresentamos nesta seção o algoritmo linear VTindifference para o problema VTP em grafos de intervalos unitários.

Algoritmo VTindifference

Entrada: Um grafo G irredundante de intervalos unitários.

- 1 Encontre uma ordenação canônica $v_1 < v_2 < \dots < v_n$ dos vértices de G
- 2 $\mathcal{A}^* \leftarrow \emptyset$
- 3 $i \leftarrow 1$
- 4 ENQUANTO $i \leq n - 2$
- 5 FAÇA SE $v_i v_{i+2} \in E_G$
- 6 ENTÃO $\{T \leftarrow [v_i, v_{i+1}, v_{i+2}], \mathcal{A}^* \leftarrow \mathcal{A}^* \cup T, G \leftarrow G - V_T\}$
- 7 $i \leftarrow i + 3$
- 8 DEVOLVA \mathcal{A}^*

Teorema 3.16 *O algoritmo VTindifference aplicado a um grafo de intervalos unitários G devolve um empacotamento máximo de $\mathcal{T}_V(G)$ em tempo linear.*

Demonstração. Suponha que, para um grafo de intervalos unitários G , a coleção \mathcal{A}^* devolvida pelo algoritmo não seja uma solução ótima. Sejam \mathcal{T}^* um empacotamento máximo de $\mathcal{T}_V(G)$ que tem o número máximo de triângulos em comum com \mathcal{A}^* , e i o menor número tal

que $[v_i, v_{i+1}, v_{i+2}] \in \mathcal{A}^* \setminus \mathcal{T}^*$. Denotamos o triângulo $[v_i, v_{i+1}, v_{i+2}]$ por T . Vamos mostrar que

$$\text{Se } T' \in \mathcal{T}^*, T' \cap T \neq \emptyset \text{ e } v_j \in V_{T'} \setminus V_T, \text{ então } j > i + 2. \quad (3.32)$$

Suponha, por contradição, que $j \leq i + 2$. Como $v_j \notin V_T$, temos que $j < i$.

Se v_j é um vértice de um triângulo T'' em \mathcal{A}^* , pela definição do número i e pelo fato de que $j < i$, temos que $T'' \in \mathcal{T}^* \cap \mathcal{A}^*$. Além disso, como $T \in \mathcal{A}^*$ e $T' \cap T \neq \emptyset$ podemos ver que $T' \notin \mathcal{A}^*$, e portanto $T' \neq T''$. Notamos que nesse caso v_j é coberto por T' e T'' , ambos triângulos de \mathcal{T}^* , o que é impossível. Segue, portanto, que v_j não é coberto por qualquer triângulo de \mathcal{A}^* . Analisamos dois possíveis casos.

- (i) $v_j v_{i+1} \in E_G$ ou $v_j v_{i+2} \in E_G$. De $j < i$ e (3.31), segue que $v_j v_{j+2} \in E_G$. Como v_j não está coberto por nenhum triângulo de \mathcal{A}^* , o algoritmo incluiria o triângulo $[v_j, v_{j+1}, v_{j+2}]$ na solução, uma contradição.
- (ii) $v_j v_{i+1}, v_j v_{i+2} \notin E_G$. Nesse caso, é claro que $v_i, v_j \in V_{T'}$. Seja v_k o terceiro vértice de T' . Note que $v_k \notin V_T$. Se $k \leq i + 2$ então (assim como para v_j) segue que $k < i$ e que v_k não está coberto por nenhum triângulo de \mathcal{A}^* . Para $l := \min\{j, k\}$, $j, k < i$, $v_l v_i \in E_G$ e (3.31) implicam em $v_l v_{l+2} \in E_G$. Mas nesse caso, o algoritmo incluiria o triângulo $[v_l, v_{l+1}, v_{l+2}]$ na solução, uma contradição. Se $k > i + 2$ então $j < i$, $v_j v_k \in E_G$ e (3.31) implicam em $v_j v_{j+2} \in E_G$, novamente uma contradição.

Provamos então a afirmação (3.32).

Se um único triângulo T_1 de \mathcal{T}^* intersecta T , substituindo T_1 por T obtemos um empacotamento máximo de $\mathcal{T}_V(G)$ que contém mais triângulos em comum com \mathcal{A}^* do que \mathcal{T}^* , uma contradição.

Se exatamente dois triângulos T_1 e T_2 , de \mathcal{T}^* intersectam T , como T_1 e T_2 são disjuntos nos vértices existem pelo menos 3 vértices distintos, digamos v_k, v_l e v_p , em $(V_{T_1} \cup V_{T_2}) \setminus V_T$. De (3.32) temos que $k, l, p > i + 2$. Suponha, sem perda de generalidade, que $p = \max\{k, l, p\}$. Como v_p é adjacente a pelo menos um dos vértices v_i, v_{i+1}, v_{i+2} , e $k, l, p > i + 2$, por (3.31) temos que $[v_k, v_l, v_p]$ é um triângulo em G . Substituindo T_1 e T_2 por T e $[v_k, v_l, v_p]$, obtemos uma solução ótima que tem mais triângulos em comum com \mathcal{A}^* do que \mathcal{T}^* , uma contradição.

Se três triângulos, T_1 , T_2 , e T_3 de \mathcal{T}^* intersectam T , então existem 6 vértices distintos em $(V_{T_1} \cup V_{T_2} \cup V_{T_3}) \setminus V_T$. Por (3.32), sabemos que os índices de todos estes triângulos são maiores que $i + 2$. Além disso, assim como fizemos acima, usando (3.31) concluímos que estes triângulos induzem um subgrafo completo de G . Portanto, podemos substituir T_1 , T_2 e T_3 por T e dois outros triângulos disjuntos nos vértices, cujos vértices estão em $(V_{T_1} \cup V_{T_2} \cup V_{T_3}) \setminus V_T$. Temos novamente uma contradição.

Como a ordenação canônica pode ser computada em tempo linear [26] e como a iteração das linhas 3 - 5 claramente consomem tempo linear, segue que o algoritmo VTindifference é linear. \square

3.4.3 Observações finais

Investigamos o problema VTP para a classe dos grafos de intervalos. Não conseguimos encontrar um algoritmo polinomial para este caso, e nem tampouco mostrar que este caso é

NP-difícil. Observamos que um algoritmo guloso simples não resolve o VTP para essa classe de grafos. Na Figura 3.19 exibimos um grafo de intervalos (não-unitários) para o qual uma solução gulosa não é uma solução ótima. Técnicas como programação dinâmica, que aparentam ser apropriadas para este caso, não nos conduziram a resultados positivos. Seria interessante obter um resultado para este caso. Lembramos que, para a classe dos grafos cordais (que contém a classe dos grafos intervalos), o problema VTP é NP-difícil, conforme mencionamos no início deste capítulo.

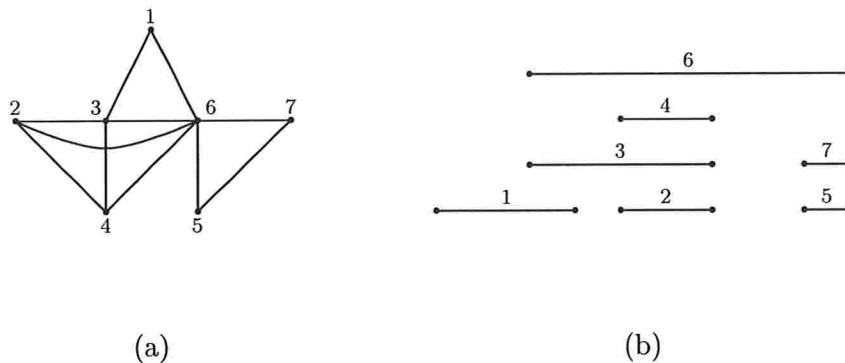


Figura 3.19: (a) Um grafo de intervalos, que é o grafo de interseção do conjunto de intervalos mostrados em (b). Um algoritmo guloso pode devolver o triângulo $[1, 3, 6]$. Uma solução ótima é a coleção $\{[2, 3, 4], [5, 6, 7]\}$.

Uma observação interessante é que o algoritmo VTindifference pode ser generalizado de modo que, dado um grafo de intervalos unitários G e um número inteiro fixo $r \geq 2$, ele encontra em G , em tempo linear, o número máximo de cliques de ordem r , duas a duas disjuntas nos vértices.

Capítulo 4

Empacotamento de triângulos disjuntos nas arestas

4.1 Introdução

Neste capítulo trataremos do problema do empacotamento máximo de triângulos disjuntos nas arestas, denotado por ETP. Holyer [21] mostrou que ETP é NP-difícil. Lembramos que ETP- k denota o problema ETP restrito a grafos de grau máximo k .

Em 2002, Caprara e Rizzi [7] exibiram um algoritmo polinomial para o problema ETP-4, e mostraram que ETP-5 é APX-difícil (isto é, não admite um esquema de aproximação polinomial a menos que $P = NP$).

Apresentamos neste capítulo um algoritmo de aproximação para o problema ETP-5 cuja razão de aproximação é $\frac{4}{3}$.

Neste capítulo, restringimos nossa atenção aos grafos irredundantes. Quando escrevemos $G - U$ (para $U \subseteq V_G$ ou $U \subseteq E_G$) assumimos que os vértices isolados e as arestas que não pertencem a nenhum triângulo foram removidos também.

4.2 Grafos arbitrários

Tal como ocorre no caso do problema VTP, um algoritmo guloso simples que retorna uma coleção maximal de triângulos disjuntos nas arestas é uma 3-aproximação para o problema ETP. Isto segue do fato de que todo triângulo de uma solução obtida pelo algoritmo guloso intersecta no máximo 3 triângulos de uma solução ótima.

Observamos também que o algoritmo $HS(\mathcal{T}, t)$ de Hurkens e Schrijver [23] para o problema do empacotamento máximo de k -conjuntos (mencionado na seção 3.2.1) nos fornece uma $(\frac{3}{2} + \epsilon)$ -aproximação para o ETP, onde ϵ é inversamente proporcional a t (conforme t aumenta, ϵ tende a zero). Analogamente ao caso do VTP, basta tomar $k = 3$ e $\mathcal{T} = \mathcal{T}_E(G)$, onde $\mathcal{T}_E(G)$ denota a coleção dos conjuntos das arestas de todos os triângulos em G . Eles exibiram ainda, para cada t , instâncias para as quais o algoritmo acima produz soluções que atingem tal razão em relação ao empacotamento máximo de triângulos disjuntos nas arestas. A razão de aproximação obtida por Hurkens e Schrijver é a melhor razão de aproximação que se conhece para o ETP (para grafos arbitrários).

4.3 Grafos com grau máximo 5

Nesta seção, restringimos nossa atenção aos grafos irredundantes de grau máximo 5.

Considerando que, em termos de grau, instâncias com grau máximo 5 para o problema ETP constituem um caso difícil, que está no limite entre instâncias difíceis e fáceis [7], investigamos o ETP-5 com o objetivo de encontrar um algoritmo com uma razão de aproximação melhor que $3/2$. Desenvolvemos um algoritmo de aproximação para o ETP-5 cuja razão de aproximação é $\frac{4}{3}$. Esse algoritmo será apresentado a seguir.

4.3.1 Algoritmo de aproximação ET5

Chamaremos de $ET5(G)$ o algoritmo de aproximação que desenvolvemos para o problema ETP-5. Neste algoritmo tem um papel fundamental o *grafo de Hajós*, denotado por $H := H[T_1, T_2, T_3]$. O grafo de Hajós consiste em um circuito de tamanho 6 e três cordas que induzem um triângulo; os outros três triângulos (cada um contendo uma destas cordas) são denotados por T_1 , T_2 e T_3 . Veja a Figura 4.1.

O algoritmo $ET5(G)$ verifica se existe um grafo de Hajós $H = H[T_1, T_2, T_3]$ em G . Se existe, ele adiciona os triângulos T_1 , T_2 e T_3 ao conjunto \mathcal{A}^* a ser devolvido pelo algoritmo, retira E_H do grafo G e repete o processo. Caso contrário, ele adiciona $\{T : E_T \in HS(\mathcal{T}_E(G), 3)\}$ ao conjunto \mathcal{A}^* (onde HS é o algoritmo baseado em busca local definido na seção 3.2.1).

Algoritmo ET5

Entrada: Um grafo G irredundante de grau máximo 5.

- 1 $\mathcal{A}^* \leftarrow \emptyset$
- 2 ENQUANTO G contém um grafo de Hajós $H = H[T_1, T_2, T_3]$
- 3 FAÇA $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup \{T_1, T_2, T_3\}$
- 4 $G \leftarrow G - E_H$
- 5 DEVOLVA $\mathcal{A}^* \cup \{T : E_T \in HS(\mathcal{T}_E(G), 3)\}$

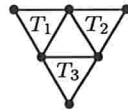


Figura 4.1: O grafo de Hajós denotado por $H[T_1, T_2, T_3]$.

4.3.2 Desempenho do algoritmo ET5

Para obter a razão de aproximação do algoritmo ET5, precisamos dos seguintes lemas.

Lema 4.1 *Se G é um grafo que contém um grafo de Hajós H , então o número de triângulos que têm alguma aresta em comum com H , em qualquer empacotamento máximo de $\mathcal{T}_E(G)$, é no máximo 4.*

A prova do lema acima pode ser obtida facilmente por inspeção.

Lema 4.2 *Se G é um grafo que não contém nenhum grafo de Hajós, o algoritmo $HS(\mathcal{T}_E(G), 3)$ é uma $\frac{4}{3}$ -aproximação para o problema ETP-5.*

Demonstração. A prova é por indução no número de triângulos em G . Para $|\mathcal{T}_G| = 0$ e $|\mathcal{T}_G| = 1$, a prova do lema é imediata. Sejam G um grafo com $|\mathcal{T}_G| > 1$, \mathcal{T}^* um empacotamento máximo de $\mathcal{T}_E(G)$, e $\mathcal{A}^* := \{T : E_T \in \text{HS}(\mathcal{T}_E(G), 3)\}$. Mostramos em seguida que existe uma coleção de triângulos \mathcal{A} tal que

$$\mathcal{A} \subseteq \mathcal{A}^*, |\mathcal{A}| \leq 3 \text{ e } \frac{|\mathcal{T}|}{|\mathcal{A}|} \leq \frac{4}{3}, \text{ onde } \mathcal{T} \text{ é um conjunto de triângulos de } \mathcal{T}^* \text{ que têm pelo menos uma aresta em comum com algum triângulo em } \mathcal{A}. \quad (4.1)$$

Para isso, analisamos quatro casos.

- (i) Se existe um triângulo $T \in \mathcal{A}^* \cap \mathcal{T}^*$, então definimos $\mathcal{A} := \{T\}$.
- (ii) Se o caso (i) não é satisfeito e \mathcal{A}^* contém triângulos distintos T_1, T_2, T_3 tais que $|V_{T_1} \cap V_{T_2}| = |V_{T_2} \cap V_{T_3}| = 1$, então definimos $\mathcal{A} := \{T_1, T_2, T_3\}$ (veja Figura 4.2). Usando o fato de que G não contém nenhum grafo de Hajós e que o grau máximo de G é 5, é fácil ver que o número de triângulos de \mathcal{T}^* que têm pelo menos uma aresta em comum com algum triângulo em \mathcal{A} é no máximo 4.

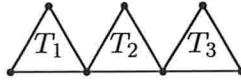


Figura 4.2: Triângulos T_1, T_2 e T_3 do caso (ii).

- (iii) Se os casos (i) e (ii) não são satisfeitos e \mathcal{A}^* contém triângulos distintos T_1 e T_2 tais que $|V_{T_1} \cap V_{T_2}| = 1$, então definimos $\mathcal{A} := \{T_1, T_2\}$. Suponha que $|\mathcal{T}| \geq 3$. Se nenhum triângulo de \mathcal{A}^* , exceto T_1 e T_2 , tem pelo menos uma aresta em comum com algum triângulo em \mathcal{T} , então T_1 e T_2 poderiam ser substituídos por \mathcal{T} , uma contradição. Portanto, existe um triângulo T_3 ($T_3 \neq T_1, T_2$) em \mathcal{A}^* que tem uma aresta em comum com algum triângulo de \mathcal{T} . Como G não contém o grafo de Hajós, temos que os triângulos T_1, T_2 e T_3 satisfazem a condição do caso (ii), o que é de novo uma contradição. Concluimos, portanto, que $|\mathcal{T}| \leq 2$.
- (iv) Se os casos (i), (ii) e (iii) não são satisfeitos, então seja T_1 um triângulo qualquer de \mathcal{A}^* e $\mathcal{A} := \{T_1\}$. Suponha que $|\mathcal{T}| \geq 2$. Se o único triângulo de \mathcal{A}^* que tem uma aresta em comum com algum triângulo em \mathcal{T} é T_1 , então T_1 poderia ser substituído por triângulos em \mathcal{T} , uma contradição. Portanto, existe um triângulo T_2 ($T_2 \neq T_1$) em \mathcal{A}^* que tem uma aresta em comum com algum triângulo de \mathcal{T} , ou seja, temos o caso (iii), uma contradição. Concluimos portanto que $|\mathcal{T}| \leq 1$.

Em todos os casos acima temos que a afirmação (4.1) é válida.

Seja \mathcal{A} uma coleção que satisfaz (4.1) e $G' := G - \bigcup_{T \in \mathcal{A}} E_T$. Observamos que $\mathcal{A}^* \setminus \mathcal{A}$ é 3-ótimo em G' (caso contrário, \mathcal{A}^* não seria 3-ótimo em G). Portanto, pela hipótese da indução temos que $\text{opt}(G') \leq \frac{4}{3}|\mathcal{A}^* \setminus \mathcal{A}|$. Logo,

$$\text{opt}(G) \leq \text{opt}(G') + |\mathcal{T}| \leq \frac{4}{3}|\mathcal{A}^* \setminus \mathcal{A}| + |\mathcal{T}| = \frac{4}{3}|\mathcal{A}^*| - \frac{4}{3}|\mathcal{A}| + |\mathcal{T}|.$$

Como $\frac{|T|}{|\mathcal{A}|} \leq \frac{4}{3}$, das desigualdades acima segue que $\text{opt}(G) \leq \frac{4}{3}|\mathcal{A}^*|$. \square

Teorema 4.3 *O algoritmo ET5 é uma $\frac{4}{3}$ -aproximação para o problema ETP-5.*

Demonstração. Seja G' o grafo G imediatamente antes da execução da linha 5 do algoritmo ET5 e j o número de iterações (linhas 2 – 4) feitas pelo algoritmo. Pelo Lema 4.1, $\text{opt}(G') \geq \text{opt}(G) - 4j$. Se $G' \neq \emptyset$, então claramente, G' não contém nenhum grafo de Hajós. Logo, pelo Lema 4.2, $\text{opt}_{\text{ETP}}(G') \leq \frac{4}{3}|\{T : E_T \in \text{HS}(\mathcal{T}_E(G'), 3)\}|$. Além disso, $|\mathcal{A}^*| = 3j$ (onde \mathcal{A}^* é o conjunto na linha 5 do algoritmo). Portanto,

$$\begin{aligned} \text{opt}_{\text{ETP}}(G) &\leq 4j + \text{opt}_{\text{ETP}}(G') \leq 4\frac{|\mathcal{A}^*|}{3} + \frac{4}{3}|\{T : E_T \in \text{HS}(\mathcal{T}_E(G'), 3)\}| \\ &= \frac{4}{3}|\mathcal{A}^* \cup \{T : E_T \in \text{HS}(\mathcal{T}_E(G'), 3)\}|. \end{aligned}$$

\square

Proposição 4.4 *A razão de aproximação $\frac{4}{3}$ do algoritmo ET5 é justa.*

Demonstração. Considere o grafo da Figura 4.3.

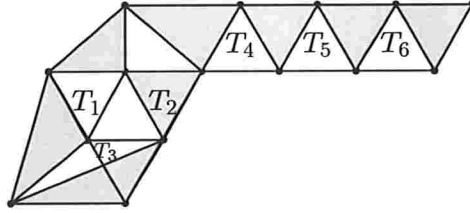


Figura 4.3: Os 8 triângulos destacados compõem uma solução ótima de ETP. O algoritmo ET5 pode devolver os triângulos T_1, \dots, T_6 (se começar removendo $H = H[T_1, T_2, T_3]$).

\square

4.3.3 Complexidade computacional do algoritmo ET5

Não é difícil ver que o algoritmo ET5 consome tempo $O(n_G^3)$, onde n_G é a ordem do grafo de entrada. De fato, as linhas 2 – 4 podem ser executadas em tempo $O(|\mathcal{T}_G|^2)$, pois em cada iteração, verifica-se para cada triângulo se ele é o centro de um grafo de Hajós. Ainda, o número de iterações é limitado por $|\mathcal{T}_G|/4$. Além disso, o conjunto $\{T : E_T \in \text{HS}(\mathcal{T}_E(G), 3)\}$ pode ser encontrado em tempo $O(|\mathcal{T}_G|^3)$, pois em cada iteração do algoritmo $\text{HS}(\mathcal{T}_E(G), 3)$, por meio de uma busca exaustiva, tentamos trocar 2 triângulos aresta-disjuntos da coleção atual por 3 triângulos aresta-disjuntos. Para isto, consumimos tempo $O(|\mathcal{T}_G|^2)$; ainda, o algoritmo $\text{HS}(\mathcal{T}_E(G), 3)$ tem no máximo $|\mathcal{T}_G|$ iterações. Finalmente, como $\Delta(G) = 5$, do Lema 3.1 temos que $|\mathcal{T}_G| \leq 4n_G$. Logo, o consumo de tempo do algoritmo ET5 é $O(n_G^3)$.

Capítulo 5

Empacotamento de arestas e triângulos

5.1 Introdução

O problema do $K_2 + K_3$ -empacotamento é o problema do \mathcal{F} -empacotamento em que $\mathcal{F} = \{K_2, K_3\}$, e o objetivo é encontrar um grafo $H \subseteq G$ que seja uma união de *grafos disjuntos nos vértices*, sendo cada um deles isomorfo a um grafo de \mathcal{F} , tal que H tenha o *maior número possível de arestas*. Observamos que se o objetivo é maximizar o número de *vértices* de H , então o problema correspondente pode ser resolvido em tempo polinomial [19]. Já o problema de nosso interesse é NP-difícil. Para verificar isso, pode-se fazer uma redução a partir do problema do empacotamento de triângulos (como fizemos na prova do Teorema 5.1).

Neste capítulo mostramos que o $K_2 + K_3$ -empacotamento é APX-completo em grafos com grau máximo 5. Para isso, mostramos uma L-redução do problema VTP em grafos de grau máximo 4. Mostramos também que o $K_2 + K_3$ -empacotamento é APX-completo mesmo em grafos de grau máximo 4 tais que toda aresta pertence a algum triângulo. Utilizamos uma L-redução do problema MAX2SAT3. Ela é mais elaborada, e é baseada na idéia de Caprara e Rizzi [7] para mostrar que o problema VTP em grafos de grau máximo 4 é APX-completo.

Definimos o problema MAX2SAT3 como segue. Dada uma coleção de cláusulas $C = \{c_1, c_2, \dots, c_l\}$ sobre um conjunto finito $X = \{x_1, x_2, \dots, x_p\}$ de variáveis, com $|c_i| = 2$ para $1 \leq i \leq l$, e tal que toda variável aparece em no máximo três cláusulas (contando-se ambas as ocorrências positivas e negativas), encontre uma atribuição às variáveis de X que satisfaz o maior número possível de cláusulas. Sabe-se que o problema MAX2SAT3 é APX-completo [4].

Ainda, mostramos uma $(1 + \frac{1}{3}\rho)$ -aproximação para o $K_2 + K_3$ -empacotamento, onde ρ é uma razão de aproximação para o problema VTP. Nosso algoritmo inicialmente aplica uma rotina para encontrar o maior número possível de triângulos disjuntos nos vértices, e depois complementa a solução com um emparelhamento máximo no grafo obtido removendo-se os vértices dos triângulos que a rotina encontrou. Usando como rotina o algoritmo de Hurkens e Schrijver [23] para o VTP, mostramos que a razão de aproximação do nosso algoritmo é $\frac{3}{2} + \varepsilon$. Ressaltamos ainda que o algoritmo de aproximação para o VTP-4 que obtivemos (com razão um pouco menor que 1,2) resulta em um algoritmo de aproximação com razão 1,4 para o problema do $K_2 + K_3$ -empacotamento em grafos de grau máximo 4.

No final deste capítulo, apresentamos uma $3/2$ -aproximação para o problema do $K_2 + K_3$ -empacotamento, sugerida recentemente por Raphael Yuster. Também comentamos que um resultado obtido por Halldórsson [16] para cobertura por 3-conjuntos fornece uma $3/2$ -aproximação para esse problema.

5.2 Resultados de inaproximabilidade

Teorema 5.1 *O problema do $K_2 + K_3$ -empacotamento é APX-completo em grafos de grau máximo 5.*

Demonstração. Exibiremos uma L-redução do problema VTP-4 em grafos irredundantes que é APX-completo [7] ao problema do $K_2 + K_3$ -empacotamento. Dado um grafo irredundante G com $\Delta(G) = 4$, definimos $G' := f(G)$ como a união de duas cópias G_1 e G_2 de G , junto com o conjunto de arestas $\{u_1u_2 : u_1 \in V_{G_1}, u_2 \in V_{G_2}, \text{ e } u_1, u_2 \text{ correspondem a um mesmo vértice } u \in V_G\}$ (veja a Figura 5.1).



Figura 5.1: Uma L-redução do problema VTP-4 em grafos irredundantes ao problema do $K_2 + K_3$ -empacotamento.

Inicialmente mostramos que

$$\text{opt}_{K_2+K_3}(G') = 3 \text{opt}_{\text{VTP}}(G) + n_G. \quad (5.1)$$

Se \mathcal{T}^* é uma solução ótima de VTP em G , então um empacotamento de $\{K_2, K_3\}$ em G' consiste em triângulos de G_1 e G_2 que são cópias de triângulos em \mathcal{T}^* , e o conjunto de arestas $\{u_1u_2 : u_1 \in V_{G_1}, u_2 \in V_{G_2}, \text{ e } u_1, u_2 \text{ correspondem a um mesmo vértice } u \text{ de } G \text{ não coberto por } \mathcal{T}^*\}$. Como o número de vértices de G não cobertos por \mathcal{T}^* é $n_G - 3 \text{opt}_{\text{VTP}}(G)$, temos $\text{opt}_{K_2+K_3}(G') \geq 3 \text{opt}_{\text{VTP}}(G) + n_G$. Por outro lado, se uma solução ótima de $K_2 + K_3$ -empacotamento em G' tem t triângulos e e arestas, como $e \leq \frac{n_{G'} - 3t}{2} = n_G - \frac{3}{2}t$, temos que $\text{opt}_{K_2+K_3}(G') = 3t + e \leq \frac{3}{2}t + n_G$. Claramente, $t \leq 2 \text{opt}_{\text{VTP}}(G)$, e portanto $\text{opt}_{K_2+K_3}(G') \leq 3 \text{opt}_{\text{VTP}}(G) + n_G$. Logo, a afirmação (5.1) é verdadeira.

Seja \mathcal{T}^* uma solução ótima de VTP em G . Suponha que exista um triângulo $T \in \mathcal{T}^*$, tal que T tenha 5 vértices vizinhos em $V_G \setminus V_T$ que não são cobertos por \mathcal{T}^* . Como $\Delta(G) = 4$, um par deles, digamos v_1, v_2 , é adjacente a um mesmo vértice, digamos x , de V_T ; outro par, digamos v_3, v_4 , é adjacente a um mesmo vértice, digamos y , de V_T . Note que o terceiro vértice de V_T , digamos z , tem grau pelo menos 3. Além disso, como G é irredundante e $\Delta(G) = 4$, temos que $v_1v_2, v_3v_4 \in E_G$. De fato, como G é irredundante, a aresta v_1x (resp. v_2x) é aresta de algum triângulo. Como $d_G(x) = d_G(y) = \Delta(G) = 4$, o único triângulo possível que tem a

aresta xv_1 mas não tem a aresta v_1v_2 , é o triângulo $[x, v_1, z]$ (veja a Figura 5.2). Mas agora, o único triângulo possível que tem a aresta xv_2 é o triângulo $[x, v_2, v_1]$, e portanto, $v_1v_2 \in E_G$. Similarmente, $v_3v_4 \in E_G$.

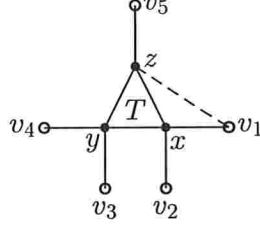


Figura 5.2: Os vértices v_1, v_2, v_3, v_4 e v_5 estão na vizinhança de V_T , e não são cobertos por \mathcal{T}^* .

Logo, substituindo T por $[x, v_1, v_2]$ e $[y, v_3, v_4]$, obtemos uma solução para VTP que tem mais triângulos do que \mathcal{T}^* , uma contradição. Portanto, todo triângulo de \mathcal{T}^* tem no máximo 4 vizinhos não cobertos por \mathcal{T}^* . Ainda, como G é irredundante, todo vértice não coberto por \mathcal{T}^* é adjacente a pelo menos um vértice coberto por \mathcal{T}^* . De fato, suponha que exista um vértice v não coberto por \mathcal{T}^* , e não adjacente a nenhum triângulo coberto por \mathcal{T}^* . Como G é irredundante, v é vértice de um triângulo T . Observe que nenhum vértice de T é coberto por \mathcal{T}^* , e portanto, \mathcal{T}^* não é solução ótima de VTP em G , uma contradição. Segue, então, que o número de vértices em G não cobertos por \mathcal{T}^* é no máximo $4 \text{opt}_{\text{VTP}}(G)$, ou seja, $n_G - 3 \text{opt}_{\text{VTP}}(G) \leq 4 \text{opt}_{\text{VTP}}(G)$. Usando (5.1), temos $\text{opt}_{K_2+K_3}(G') \leq 10 \text{opt}_{\text{VTP}}(G)$, isto é, $\alpha = 10$.

Se \mathcal{A} é um empacotamento de $\{K_2, K_3\}$ em G' , definimos $g(G, \mathcal{A})$ como o maior dos dois conjuntos $\mathcal{T}_{\mathcal{A}} \cap G_1, \mathcal{T}_{\mathcal{A}} \cap G_2$. Suponha, sem perda de generalidade, que $g(G, \mathcal{A}) = \mathcal{T}_{\mathcal{A}} \cap G_1$. Seja $t_1 := |\mathcal{T}_{\mathcal{A}} \cap G_1|$, $t_2 := |\mathcal{T}_{\mathcal{A}} \cap G_2|$, $e_1 := |\mathcal{E}_{\mathcal{A}} \cap G_1|$, $e_2 := |\mathcal{E}_{\mathcal{A}} \cap G_2|$, e seja e o número de arestas em \mathcal{A} com um extremo em G_1 e outro em G_2 . Claramente, $t_1 \leq \text{opt}_{\text{VTP}}(G)$. Logo, $\frac{1}{2}t_1 + \frac{3}{2}t_1 - 2 \text{opt}_{\text{VTP}}(G) \leq 0$. Como $t_2 \leq t_1$, temos que $\frac{1}{2}t_1 + \frac{3}{2}t_2 - 2 \text{opt}_{\text{VTP}}(G) \leq 0$, ou equivalentemente,

$$\text{opt}_{\text{VTP}}(G) - t_1 \leq 3 \text{opt}_{\text{VTP}}(G) + \left(\frac{3}{2}t_1 + \frac{3}{2}t_2 + e_1 + e_2 + e\right) - (3t_1 + 3t_2 + e_1 + e_2 + e). \quad (5.2)$$

Temos que $3t_1 + 3t_2 + 2e_1 + 2e_2 + 2e \leq n_{G'} = 2n_G$, e portanto,

$$\frac{3}{2}t_1 + \frac{3}{2}t_2 + e_1 + e_2 + e \leq n_G. \quad (5.3)$$

De (5.2) e (5.3) segue

$$\text{opt}_{\text{VTP}}(G) - t_1 \leq 3 \text{opt}_{\text{VTP}}(G) + n_G - (3t_1 + 3t_2 + e_1 + e_2 + e).$$

Agora, usando (5.1), obtemos $\text{opt}_{\text{VTP}}(G) - t_1 \leq \text{opt}_{K_2+K_3}(G') - \text{val}_{K_2+K_3}(f(G), \mathcal{A})$, ou seja, $\beta = 1$. Como o problema K_2+K_3 -empacotamento pertence à classe APX (veremos na seção 5.3 um algoritmo de aproximação para este problema), segue que ele é APX-completo. \square

Teorema 5.2 *O problema do $K_2 + K_3$ -empacotamento é APX-completo, mesmo em grafos irredundantes de grau máximo 4.*

Demonstração. Mostramos uma L-redução a partir do problema MAX2SAT3. Seja $\varphi = (C, X)$, $C = \{c_1, c_2, \dots, c_l\}$ e $X = \{x_1, x_2, \dots, x_p\}$, uma instância de MAX2SAT3. Seja m_i o número de ocorrências de x_i . Podemos assumir, sem perda de generalidade, que $m_i \geq 2$ (pois se x_i aparece somente em uma cláusula, podemos atribuir a x_i o valor que satisfaz tal cláusula). Definimos $G' := f(\varphi)$ como segue.

A cada cláusula c_j associamos uma *componente de teste* \mathcal{C}_j . A componente de teste de uma cláusula com dois literais consiste de 4 triângulos $[t_j^1, s_j^1, r_j^1]$, $[s_j^1, r_j^1, r_j^2]$, $[s_j^2, r_j^1, r_j^2]$, $[s_j^1, r_j^2, t_j^2]$ (mostrados na Figura 5.3(a)), enquanto a componente de teste associada a uma cláusula com um literal consiste de 3 triângulos $[t_j^1, s_j^1, r_j^1]$, $[s_j^1, r_j^1, r_j^2]$, $[s_j^2, r_j^1, r_j^2]$ (Figura 5.3(b)).

A cada variável x_i associamos uma *componente de atribuição* \mathcal{X}_i , conforme ilustramos na Figura 5.3(c). Esta componente consiste de $2m_i$ triângulos T_1, \dots, T_{2m_i} , onde $T_{2k-1} = [a_i^k, v_i^{k-1}, u_i^k]$ e $T_{2k} = [b_i^k, u_i^k, v_i^k]$, $k = 1, \dots, m_i$ (os índices superiores são modulo m_i). A *paridade* de T_k é a paridade de k .

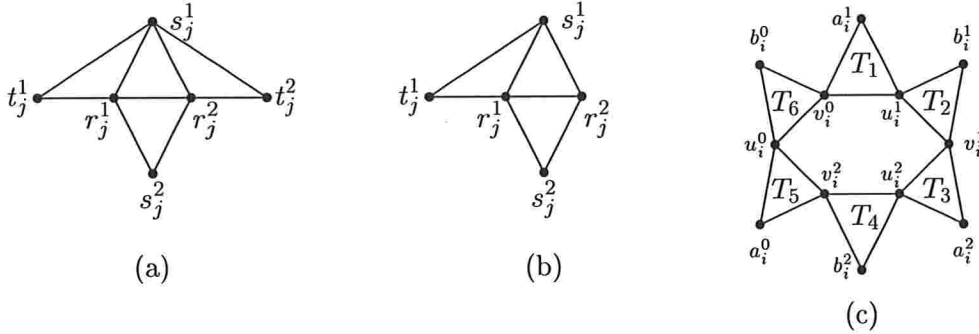


Figura 5.3: (a) Componente de teste de uma cláusula c_j com dois literais. (b) Componente de teste de uma cláusula c_j com um literal. (c) Componente de atribuição da variável x_i com $m_i = 3$.

O grafo G' é obtido conectando-se as componentes de testes e as componentes de atribuição como segue. Seja c_j uma cláusula com dois literais e sejam x_1, x_2 as variáveis que ocorrem em c_j . Se x_i ocorre positivo (resp. negado) em c_j , então identificamos o vértice t_j^i da componente de teste \mathcal{C}_j com um vértice a_i^k (resp. b_i^k) da componente de atribuição \mathcal{X}_i que ainda não foi envolvida em nenhuma identificação. Similarmente, seja c_j uma cláusula com somente um literal, digamos, x_1 . Se x_1 ocorre positivo (resp. negado) em c_j , então identificamos o vértice t_j^1 de \mathcal{C}_j , com um vértice a_1^k (resp. b_1^k) de \mathcal{X}_1 que ainda não foi envolvido em nenhuma identificação (veja a Figura 5.4). Note que G' é irredundante e $\Delta(G') = 4$.

Um empacotamento maximal \mathcal{A} de $\{K_2, K_3\}$ em G' é chamado de *canônico* se, para cada componente de atribuição, \mathcal{A} contém ou todos os triângulos pares ou todos os triângulos ímpares, e para cada componente de teste \mathcal{C}_j , \mathcal{A} contém o triângulo $[r_j^1, r_j^2, s_j^2]$, e possivelmente uma das arestas $t_j^1 s_j^1$ ou $t_j^2 s_j^1$. Inicialmente, provamos a seguinte afirmação.

Dado um empacotamento de $\{K_2, K_3\}$ não-canônico \mathcal{A} de G' , podemos construir em tempo polinomial um empacotamento canônico de G' cujo valor é pelo menos o valor de \mathcal{A} . (5.4)

Vamos construir o empacotamento desejado \mathcal{A}' a partir de \mathcal{A} . Começamos com $\mathcal{A}' = \mathcal{A}$. Inicialmente, para cada componente de atribuição \mathcal{C}_j , $1 \leq j \leq l$, removemos de \mathcal{A}' os triângulos e arestas que estão em \mathcal{C}_j e adicionamos $[r_j^1, r_j^2, s_j^2]$ a ele. Além disso, se uma das arestas $t_j^1 s_j^1$, $t_j^2 s_j^1$ é coberta por \mathcal{A} , então adicionamos a \mathcal{A}' a aresta coberta por \mathcal{A} . Observe que para cada

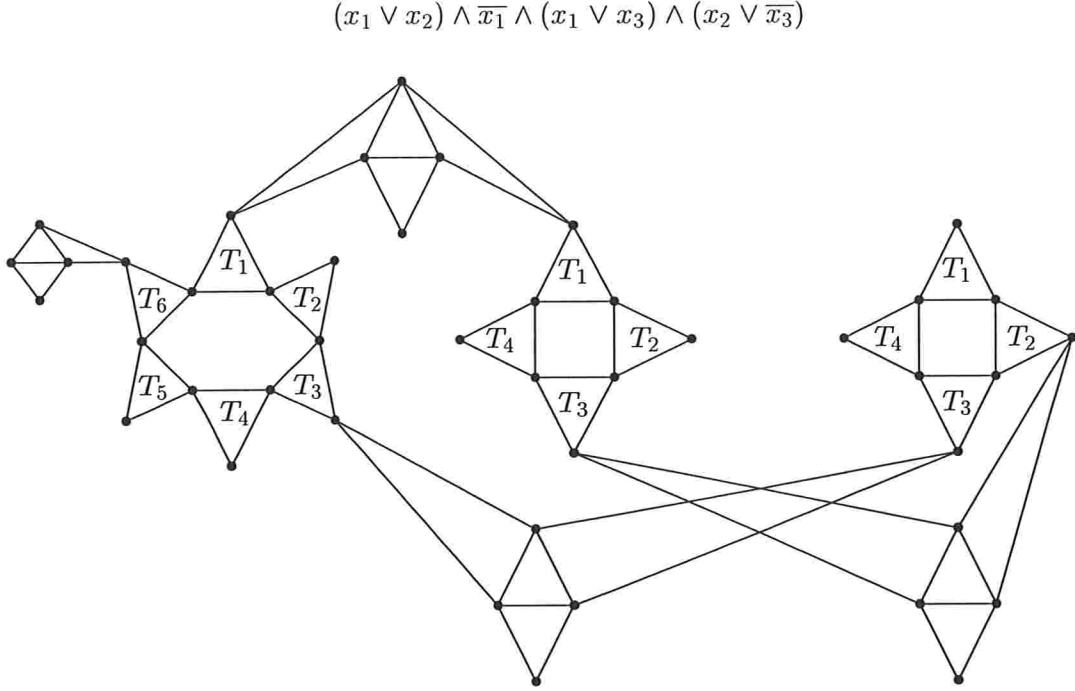


Figura 5.4: Um exemplo de construção de G' a partir de uma instância de MAX2SAT3.

\mathcal{C}_j , o valor de \mathcal{A} restrito a \mathcal{C}_j é no máximo 4. Ainda, se o valor de \mathcal{A} restrito a \mathcal{C}_j é exatamente 4, então uma das arestas $t_j^1 s_j^1$, $t_j^2 s_j^1$ é coberta por \mathcal{A} . Portanto, o empacotamento \mathcal{A}' obtido até o momento tem um valor que é pelo menos o valor de \mathcal{A} .

Além disso, para cada i , $1 \leq i \leq p$, se os triângulos da componente de atribuição \mathcal{X}_i que estão em $\mathcal{T}_{\mathcal{A}}$ não são todos de mesma paridade, fazemos o seguinte (dependendo do número de ocorrências de x_i).

1. $m_i = 3$.

Podemos supor, sem perda de generalidade, que x_i aparece negado em uma cláusula, digamos c_j , e positivo em duas cláusulas (pois se x_i aparece somente negado ou somente positivo, podemos atribuir a x_i o valor que satisfaz todas as cláusulas nas quais ela aparece). Seja t_j^k , $k \in \{1, 2\}$, o vértice de \mathcal{C}_j que incide em \mathcal{X}_i . Então, removemos de \mathcal{A}' os triângulos e arestas que estão em \mathcal{X}_i e adicionamos todos os triângulos pares de \mathcal{X}_i em \mathcal{A}' . Ainda, se $t_j^k s_j^1$ está em \mathcal{A}' , então removemos esta aresta. Mostramos em seguida que após estas mudanças o valor de \mathcal{A}' é pelo menos o valor de \mathcal{A} .

- (a) Se \mathcal{X}_i não tem nenhum triângulo que está em $\mathcal{T}_{\mathcal{A}}$, então existem no máximo 6 arestas de \mathcal{X}_i que estão em $\mathcal{E}_{\mathcal{A}}$, uma de cada triângulo. Portanto, o valor do empacotamento diminui de no máximo 7. Como o valor do empacotamento aumenta de 9, temos que o valor de \mathcal{A}' aumenta.

- (b) Se existe exatamente um triângulo de \mathcal{X}_i que está em \mathcal{T}_A , então, existem no máximo 5 arestas de \mathcal{X}_i que estão em \mathcal{E}_A , uma de cada triângulo. Logo, o valor de \mathcal{A}' diminui de no máximo 9, e aumenta de 9.
- (c) Se existem exatamente dois triângulos de \mathcal{X}_i que estão em \mathcal{T}_A , então, existem no máximo duas arestas de \mathcal{X}_i que estão em \mathcal{E}_A (veja exemplos nas Figuras 5.5(a) e (b)). Portanto, temos que o valor do empacotamento \mathcal{A}' diminui de no máximo 9, e aumenta de 9.

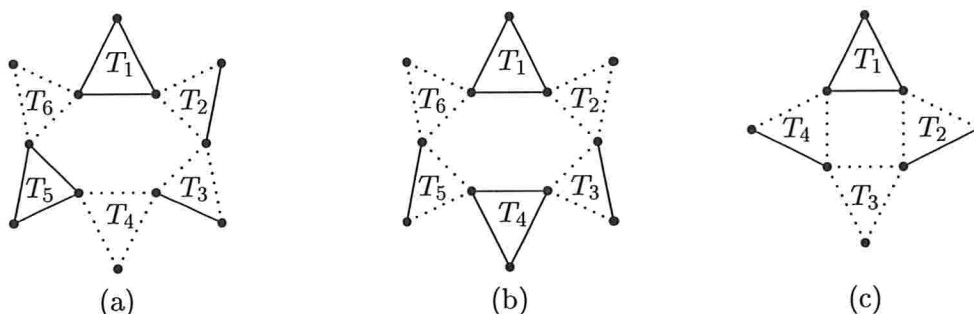


Figura 5.5: Triângulos e arestas com linhas cheias estão em \mathcal{A} . (a) $m_i = 3$ existem exatamente dois triângulos de \mathcal{X}_i de mesma paridade que estão em \mathcal{T}_A . (b) $m_i = 3$ e existem exatamente dois triângulos de \mathcal{X}_i de paridades diferentes que estão em \mathcal{T}_A . (c) $m_i = 2$ e existe um triângulo de \mathcal{X}_i que está em \mathcal{T}_A . Em todos os casos, existem no máximo 2 arestas de \mathcal{X}_i que estão em \mathcal{E}_A .

2. $m_i = 2$.

Podemos supor, sem perda de generalidade, que x_i aparece negado em uma cláusula, digamos c_j , e positivo em outra. Então, removemos de \mathcal{A}' os triângulos e as arestas que estão em \mathcal{X}_i e adicionamos dois triângulos pares de \mathcal{X}_i em \mathcal{A}' . Além disso, se $t_j^1 s_j^1$ está em \mathcal{A}' , removemos esta aresta. Mostramos em seguida que estas mudanças resultam em um empacotamento \mathcal{A}' cujo valor é pelo menos o valor de \mathcal{A} .

- (a) Se nenhum triângulo de \mathcal{X}_i está em \mathcal{T}_A , então existem no máximo 4 arestas de \mathcal{X}_i que estão em \mathcal{E}_A , uma de cada triângulo. Portanto, o valor do empacotamento \mathcal{A}' diminui de no máximo 5. Como o valor do empacotamento aumenta de 6, temos que o valor de \mathcal{A}' aumenta.
- (b) Se existe um triângulo de \mathcal{X}_i que está em \mathcal{T}_A , então existe somente um tal triângulo, digamos T_k . Além disso, existem no máximo 2 arestas de \mathcal{X}_i que estão em \mathcal{E}_A , pois o número de vértices em $\mathcal{X}_i - V_{T_k}$ é 5 (veja um exemplo na Figura 5.5(c)). Portanto, o valor de \mathcal{A}' diminui de no máximo 6, e aumenta de 6.

Finalmente, para cada componente de teste \mathcal{C}_j , se s_j^1 não é ainda o extremo de uma aresta em $\mathcal{E}_{\mathcal{A}'}$, então sempre que possível, adicionamos uma das arestas $t_j^1 s_j^1$ ou $t_j^2 s_j^1$ em \mathcal{A}' . Ou seja, se a cláusula correspondente c_j tem dois literais, então, se t_j^1 não está coberto por \mathcal{A}' , adicionamos $t_j^1 s_j^1$ em \mathcal{A}' ; caso contrário, se t_j^2 não está coberto por \mathcal{A}' , adicionamos $t_j^2 s_j^1$ a \mathcal{A}' . Se, no entanto, a cláusula c_j tem somente um literal, então se t_j^1 não está coberto por \mathcal{A}' , adicionamos $t_j^1 s_j^1$ ao \mathcal{A}' .

Na Figura 5.6 exibimos um empacotamento \mathcal{A} e o empacotamento \mathcal{A}' obtido a partir de \mathcal{A} .

Note que o empacotamento obtido \mathcal{A}' é um empacotamento canônico de G' cujo valor é pelo menos o valor de \mathcal{A} . Provamos portanto a afirmação (5.4).

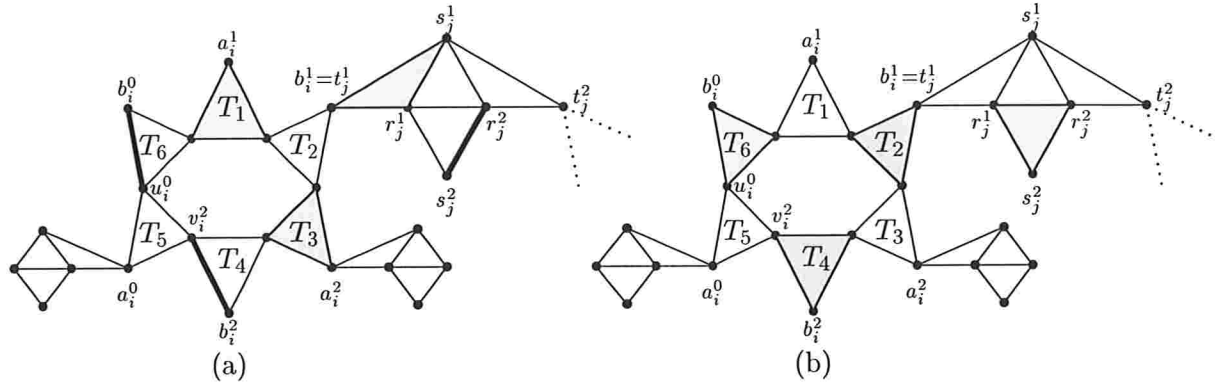


Figura 5.6: Um exemplo de construção de \mathcal{A}' (quando $m_i = 3$ e x_i aparece negado em somente uma cláusula, digamos c_j , e positivo em duas outras cláusulas). Linhas pontilhadas indicam arestas de uma outra componente de atribuição. (a) Mostra um empacotamento \mathcal{A} de $\{K_2, K_3\}$ não-canônico restrito ao \mathcal{X}_i e \mathcal{C}_j (arestas e triângulos destacados estão em \mathcal{A}). Na primeira fase, $[t_j^1, s_j^1, r_j^1]$ e $[s_j^2, r_j^2]$ são removidos de \mathcal{A}' , e $[r_j^1, r_j^2, s_j^2]$ e $[t_j^1, s_j^1]$ são adicionados a \mathcal{A}' . No segundo passo, $T_1, T_3, u_i^0, v_i^2, b_i^1, t_j^1, s_j^1$ são removidos de \mathcal{A}' e os triângulos T_2, T_4, T_6 são adicionados a \mathcal{A}' . (b) O empacotamento \mathcal{A}' obtido.

Observamos que um dado empacotamento canônico \mathcal{A}' de G' corresponde a uma atribuição (verdadeiro/falso) para as variáveis em X . De fato, se \mathcal{A}' contém todos os triângulos pares (resp. ímpares) de uma componente de atribuição \mathcal{X}_i , então x_i consideramos verdadeiro (resp. falso). Por outro lado, dada uma atribuição para X , podemos construir um empacotamento canônico \mathcal{A}' de G' da seguinte maneira. Se x_i é verdadeiro (resp. falso), adicionamos todos os triângulos pares (resp. ímpares) de \mathcal{X}_i a \mathcal{A}' . Para cada componente de teste \mathcal{C}_j , adicionamos o triângulo $[r_j^1, r_j^2, s_j^2]$ a \mathcal{A}' . Além disso, se a cláusula correspondente c_j tem dois literais, então se t_j^1 não está coberto por \mathcal{A}' adicionamos $[t_j^1, s_j^1]$ a \mathcal{A}' ; caso contrário, se t_j^2 não está coberto por \mathcal{A}' adicionamos $[t_j^2, s_j^2]$ ao empacotamento. Se, no entanto, a cláusula c_j tem somente um literal, então se t_j^1 não está coberto por \mathcal{A}' adicionamos $[t_j^1, s_j^1]$ a \mathcal{A}' (veja a Figura 5.7).

Considere agora um empacotamento canônico \mathcal{A}' e a correspondente atribuição (verdadeiro/falso) para as variáveis de X . Seja c_j uma cláusula com dois literais, e sejam x_1, x_2 as variáveis que ocorrem em c_j . Note que t_j^i (para $i = 1, 2$) não está coberto por nenhum triângulo em \mathcal{A}' que pertence ao componente da atribuição correspondente \mathcal{X}_i , se e somente se, x_i satisfaz c_j . Portanto, da construção do empacotamento canônico temos que as seguintes afirmações são equivalentes:

- a cláusula c_j é satisfatível,
- pelo menos um dos vértices t_j^1, t_j^2 não está coberto por um triângulo em \mathcal{A}' que pertence ao componente da atribuição correspondente,
- exatamente uma das arestas $[t_j^1, s_j^1], [t_j^2, s_j^2]$ está em $\mathcal{E}_{\mathcal{A}'}$,
- o valor de \mathcal{A}' restrito a \mathcal{C}_j é 4.

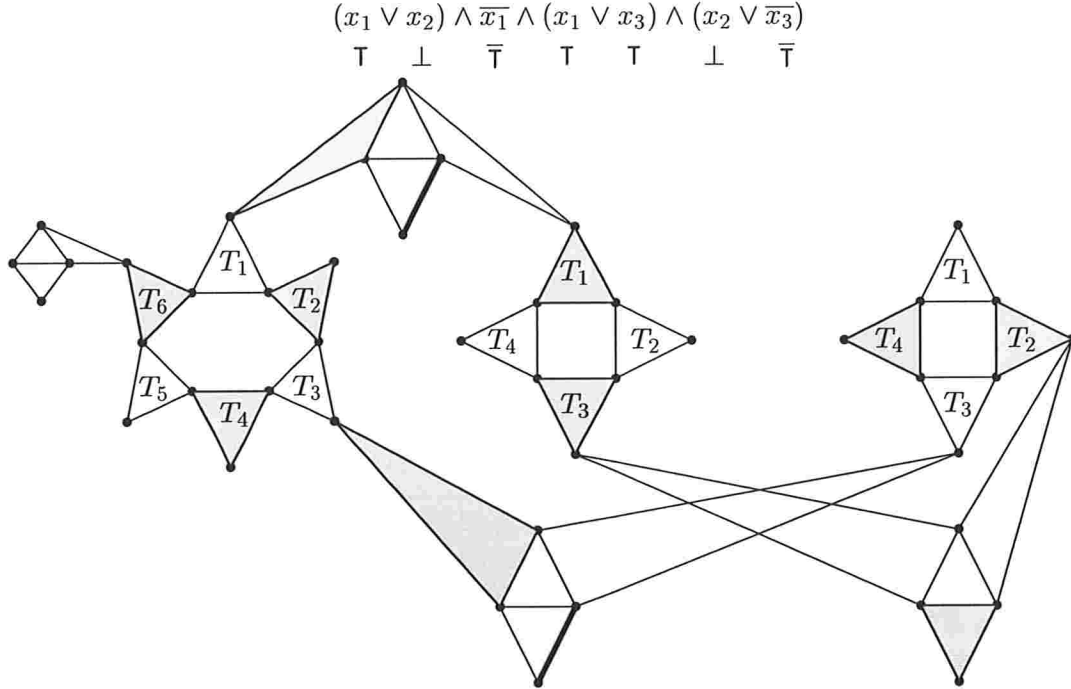


Figura 5.7: Um exemplo de uma atribuição para X e a correspondente empacotamento canônico \mathcal{A}' de G' .

Afirmações similares são verdadeiras para cláusulas com somente um literal. Portanto, o valor de \mathcal{A}' restrito a \mathcal{C}_j é 4 (resp. 3), se e somente se, c_j é satisfável (resp. não-satisfável). Além disso, exatamente m_i triângulos de cada \mathcal{X}_i estão em \mathcal{A}' . Logo, temos que a seguinte afirmação é válida.

Um empacotamento canônico \mathcal{A}' de G' com valor $\sum_{i=1}^p 3m_i + 4k + 3(l - k)$ corresponde a uma atribuição para as variáveis de X que satisfaz exatamente k cláusulas de φ , e vice-versa. (5.5)

Agora, dado um empacotamento \mathcal{A} de $\{K_2, K_3\}$ em $G' := f(\varphi)$, definimos uma atribuição $g(\varphi, \mathcal{A})$ do seguinte modo. Inicialmente, encontramos um empacotamento canônico \mathcal{A}' de G' cujo valor é pelo menos o valor de \mathcal{A} . Atribuímos a x_i o valor verdadeiro (resp. falso) se \mathcal{A}' contém todos os triângulos pares (resp. ímpares) da componente de atribuição \mathcal{X}_i .

Mostramos em seguida que

$$\text{opt}_{K_2+K_3}(G') = \sum_{i=1}^p 3m_i + \text{opt}_{\text{MAX2SAT3}}(\varphi) + 3l. \quad (5.6)$$

De fato, de (5.5) temos que uma solução ótima de $\text{MAX2SAT3}(\varphi)$ corresponde a um empacotamento canônico \mathcal{A}' de G' com valor $\sum_{i=1}^p 3m_i + 4 \text{opt}_{\text{MAX2SAT3}}(\varphi) + 3(l - \text{opt}_{\text{MAX2SAT3}}(\varphi))$.

Portanto, $\text{opt}_{K_2+K_3}(G') \geq \sum_{i=1}^p 3m_i + \text{opt}_{\text{MAX2SAT3}}(\varphi) + 3l$. Por outro lado, seja \mathcal{A} um $\{K_2, K_3\}$ -empacotamento de G' . Se a solução viável correspondente $g(\varphi, \mathcal{A})$ de $\text{MAX2SAT3}(\varphi)$ satisfaz k cláusulas, temos que $k \leq \text{opt}_{\text{MAX2SAT3}}(\varphi)$. Ainda, $\text{val}_{K_2+K_3}(G', \mathcal{A}) \leq \text{val}_{K_2+K_3}(G', \mathcal{A}')$, e por (5.5), $\text{val}_{K_2+K_3}(G', \mathcal{A}') = \sum_{i=1}^p 3m_i + k + 3l$. Portanto, temos que $\text{opt}_{K_2+K_3}(G') \leq \sum_{i=1}^p 3m_i + \text{opt}_{\text{MAX2SAT3}}(\varphi) + 3l$. Provamos então a afirmação (5.6).

Como cada cláusula tem no máximo 2 literais, temos que $\sum_{i=1}^p m_i \leq 2l$. Além disso, note que o valor ótimo para instância φ do problema MAX2SAT3 é pelo menos $\frac{l}{2}$, pois pelo menos metade das cláusulas pode ser satisfeita usando-se um algoritmo guloso simples. Logo, de (5.6) temos que $\text{opt}_{K_2+K_3}(G') \leq 9l + \text{opt}_{\text{MAX2SAT3}}(\varphi) \leq 19 \text{opt}_{\text{MAX2SAT3}}(\varphi)$, ou seja, $\alpha = 19$.

Finalmente, suponha que $\text{val}_{\text{MAX2SAT3}}(\varphi, g(\varphi, \mathcal{A})) = k$, isto é, que a atribuição $g(\varphi, \mathcal{A})$ satisfaz exatamente k cláusulas de φ . Portanto, por (5.5) temos $\text{val}_{K_2+K_3}(G', \mathcal{A}') = \sum_{i=1}^p 3m_i + k + 3l$. Desta afirmação, da igualdade (5.6), e do fato de que $\text{val}_{K_2+K_3}(G', \mathcal{A}') \geq \text{val}_{K_2+K_3}(G', \mathcal{A})$, temos que

$$\text{opt}_{\text{MAX2SAT3}}(\varphi) - \text{val}_{\text{MAX2SAT3}}(\varphi, g(\varphi, \mathcal{A})) \leq \text{opt}_{K_2+K_3}(G') - \text{val}_{K_2+K_3}(G', \mathcal{A}).$$

Portanto, $\beta = 1$. O resultado do teorema segue do fato de que o problema $K_2 + K_3$ -empacotamento pertence a classe APX e MAX2SAT3 é APX-completo [4]. \square

5.3 Algoritmo de aproximação

Apresentamos nesta seção um algoritmo que desenvolvemos para o problema do $K_2 + K_3$ -empacotamento, denominado K_2K_3 . Dado um grafo G , o algoritmo K_2K_3 inicialmente encontra em G uma coleção de triângulos disjuntos nos vértices, aplicando um algoritmo de aproximação A_{VTP} para o problema VTP. Em seguida, o algoritmo K_2K_3 complementa a solução com um emparelhamento máximo no grafo obtido após remover os triângulos que foram encontrados pelo algoritmo A_{VTP} .

Algoritmo K_2K_3

Entrada: Um grafo G .

- 1 $\mathcal{T}_A \leftarrow A_{\text{VTP}}(G)$
- 2 $\mathcal{E}_A \leftarrow$ um emparelhamento máximo em $G - \bigcup_{T \in \mathcal{T}_A} V_T$
- 3 $\mathcal{A} \leftarrow \mathcal{T}_A \cup \mathcal{E}_A$
- 4 DEVOLVA \mathcal{A}

Teorema 5.3 *Seja A_{VTP} uma ρ -aproximação para o problema VTP que devolve, para cada grafo de entrada G , uma coleção maximal de triângulos disjuntos nos vértices em G . Então, o algoritmo K_2K_3 é uma $(1 + \frac{1}{3}\rho)$ -aproximação para o problema do $K_2 + K_3$ -empacotamento.*

Demonstração. Seja G um grafo e \mathcal{A} a solução devolvida pelo algoritmo K_2K_3 quando aplicado a G . Seja \mathcal{O} uma solução ótima para o $K_2 + K_3$ -empacotamento de G com o maior número possível de triângulos em comum com \mathcal{A} . Usamos os seguintes números:

- t : o número de triângulos comuns a \mathcal{A} e \mathcal{O} .

- t_3 : o número de triângulos de \mathcal{A} que incidem em 3 triângulos diferentes de \mathcal{O} .
- t_2^v : o número de triângulos de \mathcal{A} que incidem em exatamente 2 triângulos diferentes de \mathcal{O} , tendo precisamente um vértice em comum com ambos.
- t_2^e : o número de triângulos de \mathcal{A} que incidem em exatamente 2 triângulos diferentes de \mathcal{O} , tendo uma aresta em comum com um deles.
- t_1^v : o número de triângulos de \mathcal{A} que incidem em exatamente 1 triângulo de \mathcal{O} , tendo precisamente um vértice em comum com ele.
- t_1^e : o número de triângulos de \mathcal{A} que incidem em exatamente 1 triângulo de \mathcal{O} , tendo precisamente uma aresta em comum com ele.
- t_0 : o número de triângulos de \mathcal{A} que não incidem em nenhum triângulo de \mathcal{O} .

Mostramos inicialmente que $t_0 = 0$. Suponha que $t_0 > 0$ e que T é um triângulo de \mathcal{A} que não incide em nenhum triângulo de \mathcal{O} . Se no máximo duas arestas de $\mathcal{E}_{\mathcal{O}}$ são adjacentes a T , então substituindo estas arestas por T , obtemos um $\{K_2, K_3\}$ -empacotamento com valor maior que \mathcal{O} , uma contradição. Portanto, existem 3 arestas de $\mathcal{E}_{\mathcal{O}}$ adjacentes a T . Removendo estas arestas e adicionando T em \mathcal{O} , obtemos uma solução ótima do $K_2 + K_3$ -empacotamento que tem mais triângulos em comum com \mathcal{A} do que \mathcal{O} , o que é novamente uma contradição. Logo, $t_0 = 0$.

Introduzimos agora números similares para os triângulos de \mathcal{O} .

- o : o número de triângulos comuns a \mathcal{O} e \mathcal{A} .
- o_3 : o número de triângulos de \mathcal{O} que incidem em 3 triângulos diferentes de \mathcal{A} .
- o_2^v : o número de triângulos de \mathcal{O} que incidem em exatamente 2 triângulos diferentes de \mathcal{A} , tendo precisamente um vértice em comum com ambos.
- o_2^e : o número de triângulos de \mathcal{O} que incidem em exatamente 2 triângulos diferentes de \mathcal{A} , tendo uma aresta em comum com um deles.
- o_1^v : o número de triângulos de \mathcal{O} que incidem em exatamente 1 triângulo de \mathcal{A} , tendo precisamente um vértice em comum com ele.
- o_1^e : o número de triângulos de \mathcal{O} que incidem em exatamente 1 triângulo de \mathcal{A} , tendo precisamente uma aresta em comum com ele.
- o_0 : o número de triângulos de \mathcal{O} que não incidem em nenhum triângulo de \mathcal{A} .

Como o algoritmo A_{VTP} devolve um empacotamento de triângulos maximal, o_0 é zero. Mostramos em seguida que

$$3t + 3t_3 + 2t_2^v + 2t_2^e + t_1^v + t_1^e = 3o + 3o_3 + 2o_2^v + 2o_2^e + o_1^v + o_1^e. \quad (5.7)$$

De fato, note que um triângulo de \mathcal{A} e um triângulo de \mathcal{O} podem ter exatamente um dos seguintes em comum: precisamente um vértice, precisamente uma aresta, ou três vértices (no caso em que um triângulo está em \mathcal{A} e \mathcal{O}). Seja I o número de incidências entre os triângulos de \mathcal{A} e \mathcal{O} , ou seja, I é o valor obtido pela contagem do número de arestas e vértices em comum aos triângulos de \mathcal{A} e \mathcal{O} (no caso em que um triângulo de \mathcal{A} e um triângulo de \mathcal{O}

têm precisamente uma aresta em comum, incrementamos de 1 o valor de I , e não contamos os vértices desta aresta). Claramente, o valor de cada lado da equação (5.7) é igual a I .

Agora, contando as arestas que estão na interseção de um triângulo em \mathcal{A} e um triângulo em \mathcal{O} , obtemos $3t + t_2^e + t_1^e = 3o + o_2^e + o_1^e$. Como $t = o$, temos

$$t_2^e + t_1^e = o_2^e + o_1^e. \quad (5.8)$$

Introduzimos agora os números que usaremos para as arestas em $\mathcal{E}_{\mathcal{O}}$.

- $e_{\mathcal{O}}^{\mathcal{T}}$: o número de arestas em $\mathcal{E}_{\mathcal{O}}$ que têm pelo menos um vértice em comum com um triângulo de \mathcal{A} .
- $e_{\mathcal{O}}$: o número de arestas em $\mathcal{E}_{\mathcal{O}}$ que não têm nenhum vértice em comum com algum triângulo de \mathcal{A} .

Note que $e_{\mathcal{O}}^{\mathcal{T}}$ é no máximo o número de vértices dos triângulos de \mathcal{A} não-cobertos por nenhum triângulo de \mathcal{O} , isto é,

$$e_{\mathcal{O}}^{\mathcal{T}} \leq 2t_1^v + t_2^v + t_1^e. \quad (5.9)$$

Seja G' o grafo obtido removendo-se todos os vértices dos triângulos de \mathcal{A} , ou seja, $G' := G - \bigcup_{T \in \mathcal{T}_{\mathcal{A}}} V_T$. Note que um emparelhamento de G' pode ser obtido tomando-se uma aresta de cada triângulo de \mathcal{O} que tenha exatamente um vértice em comum com um triângulo de \mathcal{A} , e tomando-se as arestas de $\mathcal{E}_{\mathcal{O}}$ que não têm nenhum vértice em comum com um triângulo de \mathcal{A} . Portanto, como $\mathcal{E}_{\mathcal{A}}$ é um empacotamento máximo de G' , temos que $|\mathcal{E}_{\mathcal{A}}| \geq o_1^v + e_{\mathcal{O}}$. Deste fato e da desigualdade (5.9), temos que

$$|\mathcal{E}_{\mathcal{O}}| = e_{\mathcal{O}}^{\mathcal{T}} + e_{\mathcal{O}} \leq 2t_1^v + t_2^v + t_1^e + |\mathcal{E}_{\mathcal{A}}| - o_1^v. \quad (5.10)$$

Consideramos agora a razão do valor de \mathcal{O} sobre o valor de \mathcal{A} ,

$$r = \frac{3|\mathcal{T}_{\mathcal{O}}| + |\mathcal{E}_{\mathcal{O}}|}{3|\mathcal{T}_{\mathcal{A}}| + |\mathcal{E}_{\mathcal{A}}|}.$$

Substituindo $|\mathcal{T}_{\mathcal{O}}|$ e usando (5.10), obtemos

$$r \leq \frac{3(o + o_3 + o_2^v + o_2^e + o_1^v + o_1^e) + (2t_1^v + t_2^v + t_1^e + |\mathcal{E}_{\mathcal{A}}| - o_1^v)}{3|\mathcal{T}_{\mathcal{A}}| + |\mathcal{E}_{\mathcal{A}}|}.$$

Como $|\mathcal{E}_{\mathcal{A}}| \geq 0$ e $r \geq 1$, podemos remover $|\mathcal{E}_{\mathcal{A}}|$ na última desigualdade, obtendo

$$r \leq \frac{3(o + o_3 + o_2^v + o_2^e + o_1^v + o_1^e) + (2t_1^v + t_2^v + t_1^e - o_1^v)}{3|\mathcal{T}_{\mathcal{A}}|}.$$

Usando (5.7), temos

$$\begin{aligned} r &\leq \frac{(3t + 3t_3 + 2t_2^v + 2t_2^e + t_1^v + t_1^e) + (o_2^v + o_2^e + 2o_1^v + 2o_1^e) + (2t_1^v + t_2^v + t_1^e - o_1^v)}{3|\mathcal{T}_{\mathcal{A}}|} \\ &= \frac{3t + 3t_3 + 3t_2^v + 2t_2^e + 3t_1^v + 2t_1^e + o_2^v + o_2^e + o_1^v + 2o_1^e}{3|\mathcal{T}_{\mathcal{A}}|}. \end{aligned}$$

Agora, de (5.8) temos

$$r \leq \frac{3(t + t_3 + t_2^v + t_2^e + t_1^v + t_1^e) + (o_2^v + o_1^v + o_1^e)}{3|\mathcal{T}_A|} = \frac{3|\mathcal{T}_A| + (o_2^v + o_1^v + o_1^e)}{3|\mathcal{T}_A|}.$$

Como $o_2^v + o_1^v + o_1^e \leq o + o_3 + o_2^v + o_2^e + o_1^v + o_1^e = |\mathcal{T}_O|$, temos $r \leq 1 + \frac{1}{3} \frac{|\mathcal{T}_O|}{|\mathcal{T}_A|}$. Como $|\mathcal{T}_O| \leq \text{opt}_{\text{VTP}}(G)$ e A_{VTP} é uma ρ -aproximação para o problema VTP,

$$\frac{|\mathcal{T}_O|}{|\mathcal{T}_A|} \leq \frac{\text{opt}_{\text{VTP}}(G)}{|\mathcal{T}_A|} \leq \rho, \text{ e portanto } r \leq 1 + \frac{1}{3}\rho.$$

□

Corolário 5.4 *Existe uma $(\frac{3}{2} + \varepsilon)$ -aproximação para o problema do $K_2 + K_3$ -empacotamento.*

Demonstração. Aplicando o algoritmo K_2K_3 com $A_{\text{VTP}} = \text{HS}(t)$, o resultado segue do Teorema 5.3 e do fato de que o algoritmo de Hurkens e Schrijver, $\text{HS}(t)$, é um algoritmo de $(\frac{3}{2} + \varepsilon)$ -aproximação para o problema VTP [23]. □

Corolário 5.5 *Existe um algoritmo de 1,4-aproximação para o problema do $K_2 + K_3$ -empacotamento em grafos de grau máximo 4.*

Demonstração. Segue do Teorema 5.3 e do Teorema 3.15. □

5.4 Observações finais

O algoritmo K_2K_3 tem razão de aproximação pelo menos $3/2$ quando $A_{\text{VTP}}(G) = \text{HS}(\mathcal{T}, t)$. Mais precisamente, para o grafo G mencionado na seção 3.3.5 (com 152 triângulos), que é uma instância justa para o algoritmo $\text{HS}(\mathcal{T}_V(G), 4)$, temos que o valor de $\text{opt}_{K_2+K_3}(G)$ sobre o valor da solução obtida pelo algoritmo K_2K_3 é $3/2$.

Notamos ainda que uma solução ótima para a versão do $\{K_2, K_3\}$ -empacotamento na qual o objetivo é encontrar um subgrafo H de G que seja uma união de grafos disjuntos nos vértices e tal que H tenha o maior número possível de *vértices* não é necessariamente uma boa solução para o nosso problema. Por exemplo, existem instâncias para o nosso problema para as quais a razão entre o valor de uma solução ótima e o número de arestas em um emparelhamento perfeito (solução ótima dessa nova versão) é 2. Veja a Figura 5.8.

Os resultados que apresentamos nas seções anteriores foram submetidos ao *9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems* (APPROX 2006). Dessa forma, tomamos conhecimento de um resultado de Halldórsson [16] para um problema de cobertura mínima por 3-conjuntos que fornece uma $3/2$ -aproximação para o problema do $K_2 + K_3$ -empacotamento. Além disso, Raphael Yuster, um dos pareceristas, sugeriu uma outra $3/2$ -aproximação para esse problema, cuja análise tem semelhanças com a análise que apresentamos para o nosso algoritmo.



Figura 5.8: As arestas destacadas estão em um emparelhamento perfeito. Os triângulos destacados estão em uma solução ótima do $K_2 + K_3$ -empacotamento.

Apresentamos em seguida algoritmo sugerido por Raphael Yuster. Inicialmente, o algoritmo encontra em um dado grafo G triângulos disjuntos nos vértices de forma gulosa; em seguida, o algoritmo completa a solução com um emparelhamento máximo no grafo obtido removendo os triângulos que foram encontrados na primeira fase. Para mostrar que esse algoritmo é uma $3/2$ -aproximação para o problema do $K_2 + K_3$ -empacotamento, considere uma solução \mathcal{A} devolvida pelo algoritmo de aproximação quando aplicado a um grafo G , e uma solução ótima \mathcal{O} para o $K_2 + K_3$ -empacotamento de G . Seja o_i , $i = 1, 2, 3$, o número de triângulos de \mathcal{O} que intersectam exatamente i vértices de $\mathcal{T}_{\mathcal{A}}$. Claramente,

$$o_1 + o_2 + o_3 = |\mathcal{T}_{\mathcal{O}}|. \quad (5.11)$$

Introduzimos agora os números que usaremos para as arestas em $\mathcal{E}_{\mathcal{O}}$.

- $e_{\mathcal{O}}^{\mathcal{T}}$: o número de arestas em $\mathcal{E}_{\mathcal{O}}$ que têm pelo menos um vértice em comum com um triângulo de \mathcal{A} .
- $e_{\mathcal{O}}$: o número de arestas em $\mathcal{E}_{\mathcal{O}}$ que não têm nenhum vértice em comum com os triângulos de \mathcal{A} .

Observe que $e_{\mathcal{O}}^{\mathcal{T}}$ é no máximo o número de vértices dos triângulos de \mathcal{A} não-cobertos por nenhum triângulo de \mathcal{O} , ou seja, $e_{\mathcal{O}}^{\mathcal{T}} \leq 3|\mathcal{T}_{\mathcal{A}}| - o_1 - 2o_2 - 3o_3$. Portanto,

$$e_{\mathcal{O}} \geq \max\{0, |\mathcal{E}_{\mathcal{O}}| - 3|\mathcal{T}_{\mathcal{A}}| + o_1 + 2o_2 + 3o_3\}. \quad (5.12)$$

Seja G' o grafo obtido removendo-se todos os vértices dos triângulos de \mathcal{A} , ou seja, $G' := G - \bigcup_{T \in \mathcal{T}_{\mathcal{A}}} V_T$. Note que um emparelhamento de G' pode ser obtido tomando-se uma aresta de cada triângulo de \mathcal{O} que tenha exatamente um vértice em comum com um triângulo de \mathcal{A} , e tomando-se as arestas de $\mathcal{E}_{\mathcal{O}}$ que não têm nenhum vértice em comum com um triângulo de \mathcal{A} . Portanto, como $\mathcal{E}_{\mathcal{A}}$ é um emparelhamento máximo de G' , temos que $|\mathcal{E}_{\mathcal{A}}| \geq o_1 + e_{\mathcal{O}}$. Deste fato e da desigualdade (5.12), temos que

$$|\mathcal{E}_{\mathcal{A}}| \geq o_1 + \max\{0, |\mathcal{E}_{\mathcal{O}}| - 3|\mathcal{T}_{\mathcal{A}}| + o_1 + 2o_2 + 3o_3\}.$$

Da inequação acima e do fato (5.11), temos que o algoritmo de aproximação produz uma solução cujo valor é pelo menos

$$3|\mathcal{T}_{\mathcal{A}}| + o_1 + \max\{0, |\mathcal{E}_{\mathcal{O}}| - 3|\mathcal{T}_{\mathcal{A}}| + o_1 + 2o_2 + 3o_3\} = 3|\mathcal{T}_{\mathcal{A}}| + o_1 + \max\{0, |\mathcal{E}_{\mathcal{O}}| - 3|\mathcal{T}_{\mathcal{A}}| - o_1 + o_3 + 2|\mathcal{T}_{\mathcal{O}}|\}.$$

Consideramos agora a razão entre o valor de \mathcal{O} e o valor de \mathcal{A} ,

$$r = \frac{3|\mathcal{T}_{\mathcal{O}}| + |\mathcal{E}_{\mathcal{O}}|}{3|\mathcal{T}_{\mathcal{A}}| + |\mathcal{E}_{\mathcal{A}}|}.$$

Analisamos os dois casos possíveis.

a) $|\mathcal{E}_{\mathcal{O}}| - 3|\mathcal{T}_{\mathcal{A}}| - o_1 + o_3 + 2|\mathcal{T}_{\mathcal{O}}| \geq 0$. Neste caso, temos que

$$r = \frac{3|\mathcal{T}_{\mathcal{O}}| + |\mathcal{E}_{\mathcal{O}}|}{3|\mathcal{T}_{\mathcal{A}}| + o_1 + |\mathcal{E}_{\mathcal{O}}| - 3|\mathcal{T}_{\mathcal{A}}| - o_1 + o_3 + 2|\mathcal{T}_{\mathcal{O}}|} \leq \frac{3|\mathcal{T}_{\mathcal{O}}| + |\mathcal{E}_{\mathcal{O}}|}{2|\mathcal{T}_{\mathcal{O}}| + |\mathcal{E}_{\mathcal{O}}|} \leq \frac{3}{2}.$$

b) $|\mathcal{E}_{\mathcal{O}}| - 3|\mathcal{T}_{\mathcal{A}}| - o_1 + o_3 + 2|\mathcal{T}_{\mathcal{O}}| < 0$. Neste caso, segue que $3|\mathcal{T}_{\mathcal{A}}| + o_1 > |\mathcal{E}_{\mathcal{O}}| + o_3 + 2|\mathcal{T}_{\mathcal{O}}| \geq |\mathcal{E}_{\mathcal{O}}| + 2|\mathcal{T}_{\mathcal{O}}|$. Logo,

$$r = \frac{3|\mathcal{T}_{\mathcal{O}}| + |\mathcal{E}_{\mathcal{O}}|}{3|\mathcal{T}_{\mathcal{A}}| + o_1} < \frac{3|\mathcal{T}_{\mathcal{O}}| + |\mathcal{E}_{\mathcal{O}}|}{2|\mathcal{T}_{\mathcal{O}}| + |\mathcal{E}_{\mathcal{O}}|} \leq \frac{3}{2}.$$

Observamos que a razão $3/2$ do algoritmo que descrevemos é justa, mesmo para grafos de grau máximo 4 (veja um exemplo na Figura 5.9).

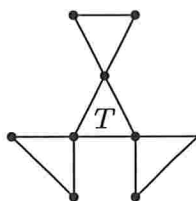


Figura 5.9: Se o algoritmo guloso de aproximação escolher inicialmente o triângulo T , então o valor devolvido pelo algoritmo é 6. O valor da solução ótima é 9.

Halldórsson [16] apresentou um algoritmo para a versão do problema de cobertura mínima por 3-conjuntos, com a condição adicional de que os conjuntos encontrados sejam dois a dois disjuntos, além de formarem uma cobertura do conjunto dos vértices do grafo. A análise do desempenho desse algoritmo baseia-se nos seguintes argumentos.

Seja \mathcal{A} a cobertura devolvida pelo algoritmo de Halldórsson e seja a_i , para $i = 1, 2, 3$, o número de conjuntos em \mathcal{A} de tamanho i . Seja \mathcal{O} uma solução ótima do problema e o_i , para $i = 1, 2, 3$, o número de conjuntos em \mathcal{O} de tamanho i . Claramente,

$$a_1 + 2a_2 + 3a_3 = b_1 + 2b_2 + 3b_3. \quad (5.13)$$

A parte central consiste em mostrar que

$$a_1 + a_2 \leq b_1 + b_2 + b_3. \quad (5.14)$$

Subtraindo (5.14) de (5.13), temos que $a_2 + 3a_3 \geq b_2 + 2b_3$, e portanto,

$$b_2 + 3b_3 \leq \frac{3}{2}(b_2 + 2b_3) \leq \frac{3}{2}(a_2 + 3a_3).$$

Ressaltamos que o algoritmo de Halldórsson é muito mais complicado do que o algoritmo proposto por Yuster. As idéias contidas na análise sugerida por Yuster guardam bastante semelhança com as idéias contidas na prova que apresentamos para o Teorema 5.3. Por outro lado, aproveitando as idéias de Yuster e juntando com outras novas, obtivemos alguns resultados preliminares interessantes para o problema do $\{K_2, \dots, K_r\}$ -empacotamento para $r > 3$. Esses resultados farão parte de um artigo a ser redigido em breve.

Capítulo 6

Considerações finais

Melhoramos a razão de aproximação para os problemas do empacotamento máximo de triângulos disjuntos nos vértices (VTP) e para o problema do empacotamento máximo de triângulos disjuntos nas arestas (ETP) em grafos de grau máximo limitado, ambos APX-completos [7]. Qualquer melhoria na razão de aproximação $\frac{3}{2} + \varepsilon$ para os problemas VTP ou ETP em grafos gerais seria muito interessante, bem como a melhoria dos limitantes inferiores para as razões de aproximação de VTP e ETP.

Uma classe de grafos para a qual não conhecemos a complexidade computacional do problema VTP é a classe dos grafos de intervalos. O problema VTP em grafos de intervalos é interessante tanto do ponto de vista prático, como do ponto de vista teórico. Como sabemos, esta classe de grafos contém a classe dos grafos de intervalos unitários, para a qual, como provamos, o problema VTP é polinomial. Também sabemos que os grafos de intervalos constituem uma subclasse dos grafos cordais para os quais VTP é NP-difícil [15].

O algoritmo de aproximação que apresentamos para o problema do $K_2 + K_3$ -empacotamento usa uma rotina para encontrar uma solução aproximada para o problema VTP. Do nosso resultado segue que qualquer melhora na razão de aproximação ($\frac{3}{2} + \varepsilon$) para o problema VTP implica em uma melhor razão de aproximação para o problema do $K_2 + K_3$ -empacotamento. Seria interessante desenvolver outras abordagens para lidar com este problema que levem a melhores razões de aproximação, ou mostrar que isto é impossível.

Um outro problema que tem despertado nosso interesse é o problema do \mathcal{F} -empacotamento no caso em que $\mathcal{F} = \{K_2, K_3, \dots, K_r\}$, para um r fixo, e o objetivo é encontrar $H \subseteq G$ que seja uma união de *grafos disjuntos nos vértices*, cada um dos quais é isomorfo a um grafo de \mathcal{F} , e tal que H tenha o *maior número possível de arestas*. Obtivemos alguns resultados de aproximação para este problema, que estão agora sendo refinados.

Recentemente, tomamos conhecimento de um problema de natureza semelhante, denominado de *partição mínima em cliques* cuja versão de otimização é a seguinte: dado um grafo $G = (V, E)$, encontre o menor k tal que existe uma partição (V_1, V_2, \dots, V_k) de V tal que, para cada $i \in \{1, 2, \dots, k\}$, o subgrafo $G[V_i]$ é uma clique. Cerioli et al. [8] provaram que este problema é MAX SNP-difícil para grafos cúbicos, e exibiram uma $5/4$ -aproximação para esta classe de grafos.

Encerramos este trabalho mencionando que há várias questões sobre problemas de \mathcal{F} -empacotamento, algumas pouco exploradas, e muitas ainda totalmente inexploradas.

Referências Bibliográficas

- [1] S. Arora. Reductions, codes, PCPs, and inapproximability. In *Proc. 36th IEEE Symp. on Foundations of Computer Science*, 1995.
- [2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *J. Assoc. Comput. Mach.*, 45(3):501–555, 1998.
- [3] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. *J. Assoc. Comput. Mach.*, 45(1):70–122, 1998.
- [4] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Prota. *Complexity and Approximation (Combinatorial Optimization Problems and Their Approximability Properties)*. Springer-Verlag, 1999.
- [5] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. Assoc. Comput. Mach.*, 41(1):153–180, 1994.
- [6] P. Berman and T. Fujito. On approximation properties of the independent set problem for low degree graphs. *Theory Comput. Syst.*, 32(2):115–132, 1999.
- [7] A. Caprara and R. Rizzi. Packing triangles in bounded degree graphs. *Inform. Process. Lett.*, 84(4):175–180, 2002.
- [8] M. R. Cerioli, L. Faria, T. O. Ferreira, C. A. J. Martinhon, F. Protti, and B. Reed. Partition into cliques for cubic graphs: planar case, complexity and approximation algorithms. Manuscrito submetido.
- [9] M. Chlebík and J. Chlebíková. Inapproximability results for bounded variants of optimization problems. In *Lecture Notes in Computer Science 2751*, pages 27–38. Springer, 2003.
- [10] M. Chlebík and J. Chlebíková. On approximability of the independent set problem for low degree graphs. In *Lecture Notes in Computer Science 3104*, pages 47–56. Springer, 2004.
- [11] G. Cornuéjols, D. Hartvigsen, and W. Pulleyblank. Packing subgraphs in a graph. *Oper. Res. Lett.*, 1(4):139–143, 1981/82.
- [12] G. Cornuéjols, D. Hartvigsen, and W. Pulleyblank. Packing subgraphs in a graph. *Oper. Res. Lett.*, 4:139–143, 1982.

- [13] P. Erdős. On even subgraphs of graphs. *Mat. Lapok*, 18:283–288, 1967.
- [14] R. L. Graham. Bounds for certain multiprocessor anomalies. *Bell System Technical Journal*, 45:1563–1581, 1966.
- [15] V. Guruswami, C. Pandu Rangan, M. S. Chang, G. J. Chang, and C. K. Wong. The K_r -packing problem. *Computing*, 66(1):79–89, 2001.
- [16] M. Halldórsson. Approximating k -set cover and complementary graph coloring. In *Integer programming and combinatorial optimization*, volume 1084 of *Lecture Notes in Computer Science*, pages 118–131. Springer, Berlin, 1996.
- [17] P. Hell and D.G. Kirkpatrick. Star factors and star packings. Technical Report 82-6, School of Computing Science, Simon Fraser University, 1982.
- [18] P. Hell and D.G. Kirkpatrick. On the complexity of general graph factor problems. *SIAM J. Comput.*, 12(3):601–609, 1983.
- [19] P. Hell and D.G. Kirkpatrick. Packings by cliques and by finite families of graphs. *Discrete Math.*, 49:45–59, 1986.
- [20] D. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Company, 1997.
- [21] I. Holyer. The NP-completeness of some edge-partition problems. *SIAM J. Comput.*, 10(4):713–717, 1981.
- [22] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *J. Algorithms*, 26(2):238–274, 1998.
- [23] C. A. J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM J. Discrete Math.*, 2(1):68–72, 1989.
- [24] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.
- [25] R. M. Karp. On the computational complexity of combinatorial problems. *Networks*, 5(1):45–68, 1975.
- [26] P. J. Looges and S. Olariu. Optimal greedy algorithms for indifference graphs. *Comput. Math. Appl.*, 25(7):15–25, 1993.
- [27] G. Manić and Y. Wakabayashi. Packing triangles in low degree graphs and indifference graphs. *Discrete Mathematics and Theoretical Computer Science, Vol. AE 2005*, pages 251–256, 2005. Disponível em http://www.ime.usp.br/~gocam/tr_pack.ps; versão completa submetida.

- [28] F. S. Roberts. Indifference graphs. In *Proof Techniques in Graph Theory*, pages 139–146. Academic Press, New York, 1969.
- [29] V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.
- [30] G. Yu and O. Goldschmidt. Local optimality and its application on independent sets for k -claw free graphs. *Journal of Combinatorial Optimization*, 1:151–164, 1997.

Índice Remissivo

- B_H , 21
- $G - U$, 5
- $G[U]$, 5
- $[u, v, w]$, 7
- $\Gamma_G(U)$, 5
- Sol_H , 21
- \mathcal{T}_G , 7
- $\mathcal{T}_E(G)$, 7
- $\mathcal{T}_V(G)$, 7
- \mathcal{T}_A , 7
- $d_G(T)$, 7
- $v_{T_1 T_2}$, 17
- \mathcal{E}_A , 7

- adjacentes, 5
- algoritmo
 - ET5, 39
 - VT4 $_k$, 21
 - VTindifference, 35
 - $K_2 K_3$, 50
 - HS(\mathcal{T} , t), 10
- aproximação
 - algoritmo de, 7
 - esquema polinomial, PTAS, 7
 - garantia, 7
 - razão, 7
- aresta, 5
- arestas
 - adjacentes, 5

- borboleta, 17

- caminho, 6
- circuito, 6
- clique, 6
- cobertura, 7
- cografo, 6

- componente, 6
 - de atribuição, \mathcal{X}_i , 45
 - de teste, \mathcal{C}_j , 45
- corda, 6

- empacotamento
 - t -ótimo, 10
 - canônico, 45
 - de $\{K_2, K_3\}$, 7
 - valor de empacotamento de $\{K_2, K_3\}$, 7
- emparelhamento, 5
- estável, 5

- grafo, 5
 - bipartido, 6
 - bipartido completo, 6
 - completo, 5
 - conexo, 6
 - cordal, 6
 - de intervalos, 6
 - de Hajós, 39
 - de interseção, 6
 - de intervalos unitários, 6
 - disco-unitário de λ -precisão, 6
 - indiferença, 6
 - irredundante, 7
 - livre de k -garras, 6
 - split, 6
 - vazio, 5
- grau
 - de T , $d_G(T)$, 7
 - de v , $d_G(v)$, 5
 - máximo de G , $\Delta(G)$, 5

- incide em, 5
- independente, 5

- interseção, $G \cap H$, 5
- localmente ótimo, 17
- ordem de G , $|G|$, 5
- ordenação canônica, 35
- Problema
 - de cobertura mínima por k -conjuntos, k -SC, 7
- problema
 - APX-completo, 8
 - APX-difícil, 8
 - MAX2SAT3, 42
 - de Conjunto Independente Máximo, MIS, 6
 - de \mathcal{F} -empacotamento, 6
 - de Empacotamento Máximo de Triângulos Disjuntos nas Arestas, ETP, ETP- k , 6
 - de Empacotamento Máximo de Triângulos Disjuntos nos Vértices, VTP, VTP- k , 6
 - do $K_2 + K_3$ -empacotamento, 7
 - em APX, 7
- procedimento
 - Aceita*(\mathcal{T}), 21
 - Adota*(H), 21
 - Redução*(H), 21
 - Restauração*(H), 22
- redução
 - AP, 8
 - L, 8
- sistema de representantes distintos, SRD, t -SRD, 7
- subgrafo, 5
 - induzido, 5
- triângulo, 5
- união, $G \cup H$, 5
- vértice, 5
 - isolado, 5
 - saturado, 16
- vizinhança de U , $\Gamma_G(U)$, 5
- vizinhos, 5