Investigando Mapeamentos entre Árvores Livres e Grafos Gêmeos

Davi Fiorese Bissoli
Marcia Helena Moreira Paiva
LabTel (Laboratório de Telecomunicações), Universidade Federal do Espírito Santo
Vitória - Espírito Santo, 29075-910, Brasil
davi.bissoli@aluno.ufes.br, marcia.paiva@ufes.br

RESUMO

Muitos problemas da vida real podem ser modelados por grafos. Esses problemas muitas vezes são classificados como NP-completos, e geralmente não há algoritmos eficientes para a solução deles. Uma maneira de superar essa dificuldade é resolver esses problemas restringindo-os a grafos especiais. As árvores formam uma classe de grafos para a qual muitos problemas NP-completos podem ser resolvidos em tempo polinomial. Neste trabalho, investigamos uma relação entre grafos gêmeos e arvores livres. Nosso objetivo é, com base nessa relação, propor técnicas para o seu mapeamento possibilitando então o desenvolvimento de métodos mais eficientes para a análise e o projeto de novas redes telecomunicações ou o aprimoramento de redes existentes. A existência de árvores livres associadas a grafos gêmeos indica a possibilidade de criação de algoritmos eficientes para o tratamento desses grafos, como algoritmos de roteamento, por exemplo. Este trabalho foi desenvolvido e redigido pelo aluno, sob supervisão da orientadora.

PALAVRAS CHAVE. Grafos gêmeos. Árvores livres. Decomposição em árvores.

Teoria e Algoritmos em Grafos. PO em Telecomunicações e Sistemas de Informações.

ABSTRACT

Many real life problems can be modelled by graphs. These problems are often classified as NP-complete, and usually there is no efficient algorithms to solve them. One way to overcome this difficulty is to solve these problems restricting them to special graphs. The trees form a class of graphs for which many NP-complete problems can be solved in polynomial time. In this work, we investigate one relationship between twin graphs and free trees. Our goal is, based on this relationship, to propose techniques for mapping it, thus enabling the development of more efficient methods for the analysis and design of new telecommunication networks or the improvement of existing ones. The existence of free trees associated with twin graph indicates the possibility of developing efficient algorithms for the treatment of these graphs, such as routing algorithms, for instance. This work was developed and written by the student, under supervision of the advisor.

KEYWORDS. Twin graphs. Free trees. Tree decomposition.

Theory and Algorithms in Graphs. OR in Telecommunications and Information Systems

1. Introdução

As redes de telecomunicações atuais estão experimentando um aumento crescente na demanda por capacidade, devido principalmente à disseminação do uso da internet. Para suportar tamanha demanda estão sendo utilizados cabos de fibra óptica na transmissão desses dados, o que chamamos de redes ópticas, as quais podem ser modeladas como grafos tendo como vértices os roteadores e como arestas as fibras ópticas.

Em [Paiva et al., 2013] foi proposto o uso de grafos gêmeos, uma classe particular de grafos 2-conexos definida em [Farley e Proskurowski, 1997], para a modelagem dessas redes. Essa classe apresenta propriedades interessantes para a área de telecomunicações, tais como um ótimo custo-benefício e alta tolerância a falhas, preservando distâncias na rede remanescente após uma falha em algum roteador. Mais sobre essa classe de grafos será explorada à frente.

Neste trabalho, investigaremos uma relação entre grafos gêmeos e árvores livres. Nosso objetivo é, a partir dessa relação, propor técnicas para o seu mapeamento possibilitando então a criação de métodos mais eficientes para melhorar a análise e o desenvolvimento de novas redes telecomunicações ou o aprimoramento de redes existentes. A existência de uma árvore livre asociada a um dado grafo gêmeo indica a possibilidade de criação de algoritmos eficientes para o tratamento desses grafos, como algoritmos de roteamento, por exemplo. Será levada em consideração a *treewidth* da árvore livre resultante, um invariante que indica a complexidade de tal algoritmo.

Em [Farley e Proskurowski, 1997] é feita a investigação de uma classe de grafos chamados auto-reparadores, da qual os grafos gêmeos fazem parte. Em meio às constatações feitas no referido artigo, é mencionada porém não explicitada a existência de uma bijeção entre grafos gêmeos e árvores livres. O interesse em tal bijeção deu a ideia para o passo inicial nas investigações realizadas no presente trabalho.

O restante do trabalho está organizado da seguinte forma: a Seção 2 apresenta conceitos sobre grafos e decomposição em árvores necessários para a compreensão do trabalho, a Seção 3 apresenta os métodos de mapeamento propostos e por último a Seção 4 conclui o trabalho.

2. Conceitos

Seja $G = G(V, E)$ um grafo G com um conjunto $V = (v_1, v_2, v_3, ...)$ de vértices ou nós e um conjunto $E = (e_1, e_2, e_3, ...)$ de arestas ou *links*, cada aresta v_iv_j conecta um par de vértices $(v_i, v_j) \in G$. Um grafo é n-conexo se existem pelo menos n caminhos disjuntos entre quaisquer dois vértices (v_i, v_j) diferentes. Se não há nenhum caminho entre pelo menos um par de vértices então o grafo é chamado desconexo. Temos também que o número de arestas incidentes a um vértice v_i é denominado o grau daquele vértice e denotado por $deg(v_i)$.

Estamos levando em consideração grafos simples, que são grafos que não possuem arestas orientadas, não possuem laços, isto é, arestas $e_j = (v_i, v_i)$, e possuem no máximo uma aresta ligando dois vértices (v_i, v_j) diferentes.

A distância d entre dois vértices (v_i, v_j) é o número de arestas de uma geodésica, ou seja, um menor caminho, conectando os dois vértices. Dizemos então que dois vértices que possuem uma distância $d = 1$ são adjacentes. Um conjunto contendo todos os vértices adjacentes a um vértice v_i é chamado de vizinhança desse vértice e denotado por $N(v_i)$.

Uma árvore é um grafo conexo e acíclico, isto é, que não possui ciclos, que são cadeias simples e fechadas. Já uma árvore livre é denominada assim se não possui um vértice raiz. Neste trabalho chamaremos árvores livres apenas de árvores.

2.1. Grafos gêmeos

Seja $G = G(V, E)$ um grafo com um conjunto V de vértices e um conjunto E de arestas. Se dois vértices (v_i, v_j) possuem a mesma vizinhança então dizemos que (v_i, v_j) é um par de gêmeos. Todo vértice de grau $deg(v) > 2$ possui um par gêmeo único, isto é, ele sempre possuirá um e somente um vértice com vizinhança idêntica à dele [Farley e Proskurowski, 1997].

Um grafo é 2-conexo se e somente se há pelo menos 2 caminhos disjuntos por vértices entre cada par de vértices $(v_i, v_j) \in G$. Como caminhos disjuntos por vértices são também disjuntos

por arestas, uma rede modelada por um grafo 2-conexo suporta falhas únicas de aresta ou vértice, pois em cada caso cada par de vértices ainda possui um caminho conectado que evita a falha. No entando, não há limitações quanto o tamanho desse caminho substituto, com respeito ao tamanho da geodésica do caminho original. Nesse respeito, a classe de grafos 2-geodesicamente-conexos (2-GC) foi definida como um subconjunto de grafos 2-conexos. Um grafo G é 2-GC se e apenas se tem pelo menos 2 geodésicas disjuntas por vértices entre cada par de vértices não adjacentes $(v_i, v_j) \in G$. Então, no caso de uma falha em algum vértice, todos os pares de vértices não adjacentes ainda estarão conectados por uma geodésica alternativa evitando a falha [Paiva et al., 2013]. Então definimos os grafos gêmeos como grafos 2-GC minimais, isto é, com o menor número de arestas possível.

Para melhor entendimento dos métodos que serão definidos na próxima seção cabe definir anteriormente outro método, esse sendo para a construção de um grafo gêmeo.

Se G é um grafo gêmeo, então o grafo G' construído pela conexão de um novo vértice por duas arestas a um par de gêmeos em G também é um grafo gêmeo. O par de gêmeos utilizado na conexão desse novo vértice será chamado de par gerador desse vértice.

Se tomarmos inicialmente o grafo G como o gêmeo mais simples, o ciclo de 4 vértices, qualquer outro grafo gêmeo pode ser construído pelo método acima [Chang et al., 1996].

2.2. Decomposição em árvores e *treewidth*

Para o tratamento da complexidade de algoritmos leva-se em consideração o conceito de *treewidth* de um grafo que é comumente usado como parâmetro na análise de complexidade de algoritmos para grafos e é definida com base em uma decomposição em árvore [Robertson e Seymour, 1986].

Uma decomposição em árvore de um grafo $G = (V, E)$ consiste em uma árvore T e um subconjunto $V_t \subseteq V$ associado a cada vértice $t \in T$. Chamaremos os subconjuntos V_t de pedaços da decomposição em árvore. T e $\{V_t : t \in T\}$ devem satisfazer:

1. Cada vértice de G pertence a pelo menos um pedaço de V_t;
2. Para cada aresta e de G, tem algum pedaço V_t contendo ambas as extremidades de e;
3. Sejam t_1, t_2 e t_3 três vértices de T tal que t_2 esteja no caminho entre t_1 e t_3. Então, se um vértice v de G pertence a ambos V_{t_1} e V_{t_3}, ele também pertence a V_{t_2}.

A Figura 1 mostra um exemplo dessa decomposição [1].

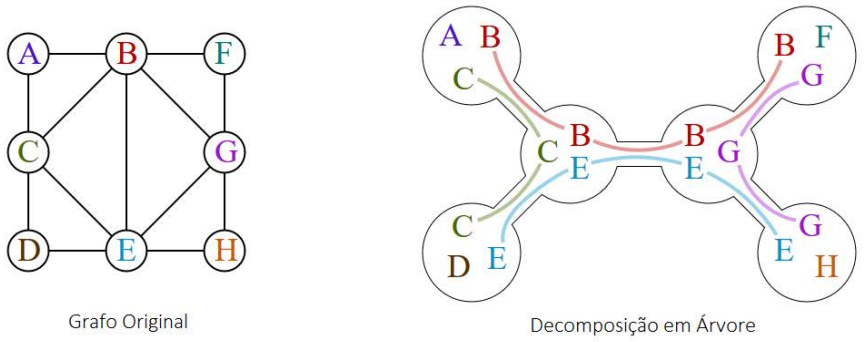

Figura 1: Exemplo de decomposição em árvore.

[1] Imagem disponível em $< https://en.wikipedia.org/wiki/File:Tree_decomposition.svg >$.

A partir da decomposição em árvore temos uma propriedade mencionada anteriormente chamada *treewidth* que é definida como o número máximo de vértices em um pedaço da decomposição em árvore menos 1 como descrito pela equação [Robertson e Seymour, 1986]:

$$treewidth(T) = max_t|V_t| - 1. \tag{1}$$

3. Métodos de mapeamento

Temos uma relação que permite o mapeamento entre conjuntos de árvores livres com m vértices internos e k folhas e o conjunto de grafos gêmeos com k vértices de grau 2 e m pares de gêmeos de grau maior.

Dado um grafo gêmeo, a relação determina uma decomposição em árvore com no máximo 4 vértices por nó:

1. Cada vértice de grau 2 está em um nó junto com seu par de gêmeos.
2. Cada par de gêmeos de grau maior está em um vértice junto com o par de gêmeos em que foi conectado no processo de construção do grafo.

Em [Farley e Proskurowski, 1997] essa relação é dita como sendo uma bijeção. Porém, ao analisar o tamanho das classes de grafos que estamos considerando (ver Tabela 1), é fácil perceber que existem muito mais árvores do que grafos 2-GC minimais. Portanto, essa relação não é de fato uma bijeção visto que, para isso, seria necessário que cada elemento da classe grafo gêmeo possuísse apenas uma árvore equivalente.

O número de árvores e de grafos gêmeos com n vértices, para $4 \leq n \leq 17$, é mostrado na Tabela 1. Os dados para as árvores estão disponíveis em [OEIS, 2010], e os dados para os grafos gêmeos podem ser encontrados em [Paiva et al., 2013].

n	árvores	grafos gêmeos
4	2	1
5	3	1
6	6	2
7	11	2
8	23	4
9	47	5
10	106	9
11	235	13
12	551	23
13	1301	35
14	3159	63
15	7741	102
16	19320	182
17	48629	310

Tabela 1: Quantidade de árvores e de grafos gêmeos com 4 a 17 vértices.

3.1. Método 1: Grafo gêmeo para árvore

Aqui será definido o processo de mapeamento de um grafo gêmeo como uma árvore.

Ao início do processo escolhemos um par de gêmeos qualquer do grafo. Independentemente do par de gêmeos escolhido, a árvore resultado será a mesma, e esse par de gêmeos ficará sozinho em um vértice da árvore resultante. Chamaremos esse par de gêmeos de gerador original.

A segunda etapa será a escolha de um vértice adjacente ao par gerador. Se esse vértice for de grau 2, criaremos uma folha na árvore resultante ligada ao vértice contendo seu par de gêmeos

gerador. Nomearemos essa folha com o vértice escolhido e o seu par de gêmeos gerador. Se esse vértice for de grau maior, será criado um vértice na árvore resultante adjacente ao vértice contendo seu par gerador. Nomearemos esse vértice com o vértice escolhido, seu par gêmeo e seu par gerador.

A segunda etapa é entao repetida para todos os vértices adjacentes aos pares de gêmeos, transformando os vértices de grau 2 em folhas na árvore e os de grau maior, unindo os pares gêmeos em um só vértice junto com seu gerador, e repetindo o processo até que todos os vértices do grafo original tenham sido convertidos para a árvore.

Esse processo pode ser feito inversamente para de uma árvore achar seu grafo gêmeo equivalente.

O fluxograma ilustrado na Figura 2 mostra o processo descrito acima. A Figura 3 mostra um exemplo simples de uma decomposição de um grafo gêmeo e sua árvore, obtida com a aplicação desse processo.

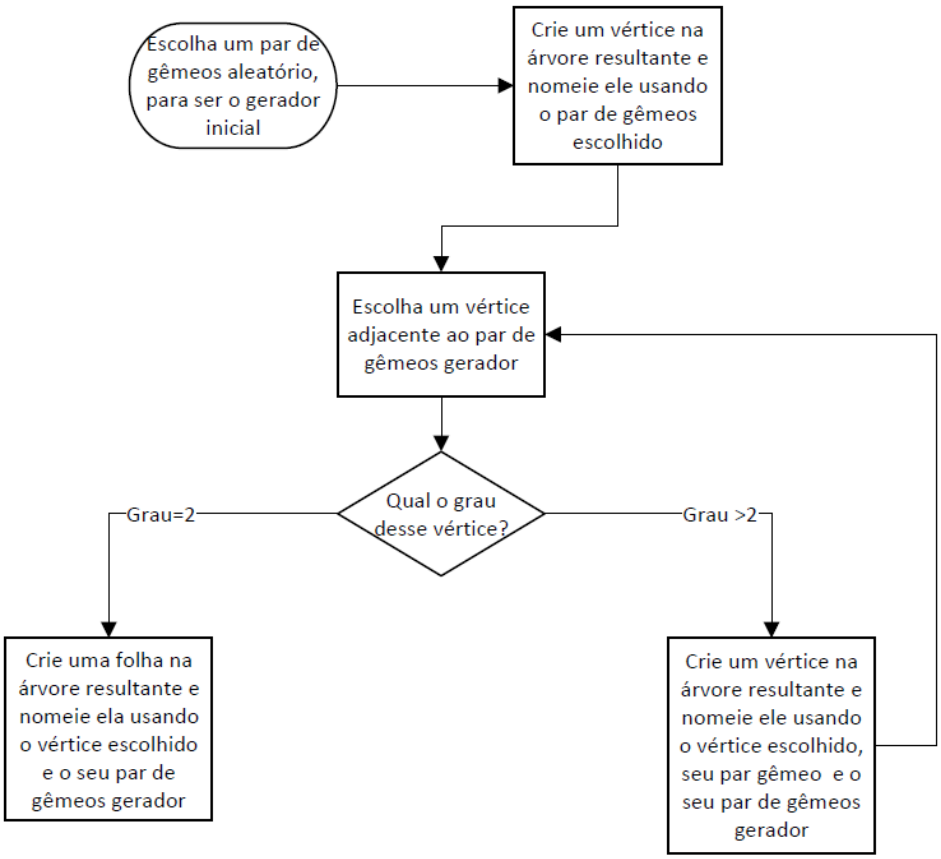

Figura 2: Fluxograma do Método 1, que mapeia grafos gêmeos em árvores.

Podemos perceber que a classe dos grafos gêmeos possui *treewidth* limitada a 3, visto que o número máximo de vértices na decomposição em árvore corresponde aos vértices que possuem dentro de si dois pares gêmeos. De fato, a maioria dos grafos gêmeos possuem árvores equivalentes com *treewidth* 3, exceto os grafos gêmeos do tipo estrela dupla, isto é, grafos bipartidos completos $K_{2,n}$, os quais possuem *treewidth* 2, pois tem apenas um par de gêmeos no grafo inicial.

3.2. Método 2: Árvore para grafo gêmeo

Se o Método 1 apresentado na seção anterior for usado inversamente, é possível encontrar um grafo gêmeo que seja equivalente a uma dada árvore livre. Porém, foi desenvolvido outro método, mais eficiente, para fazer o retorno de uma árvore a grafo gêmeo.

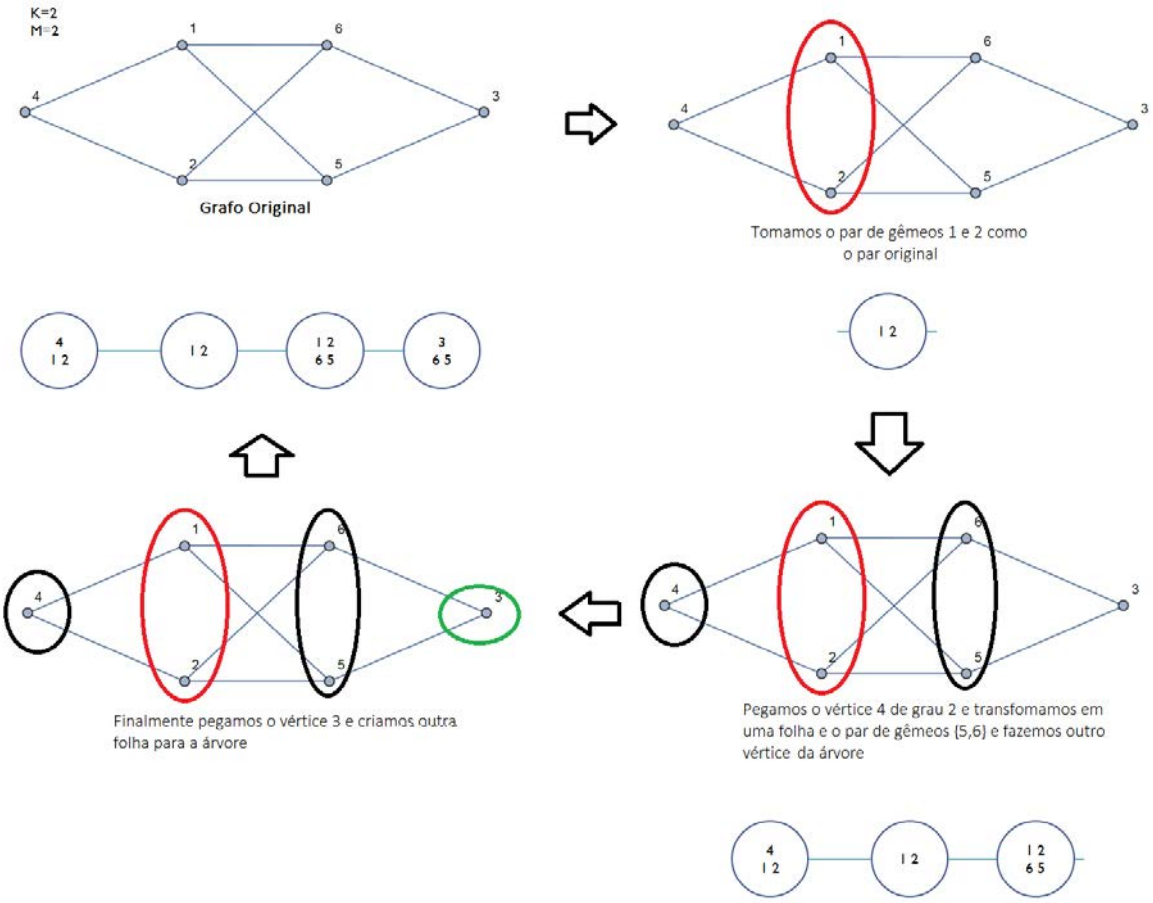

Figura 3: Exemplo de mapeamento de um grafo gêmeos em uma árvore.

Em [Nieminen e Peltola, 1998] foi desenvolvido um método para a geração de grafos k-geodesicamente conexos a partir de árvores utilizando grafos desconexos no processo. Foi necessário uma adaptação desse método para conseguir o resultado desejado, isto é, obter grafos 2-geodesicamente-conexos minimais. O método de Nieminem e Peltola tinha como resultado grafos com qualquer número de arestas, o que não apresenta grandes interesses na modelagem de redes ópticas, visto que ao aumentar o número de arestas a proteção contra falhas ainda existe porém perde-se outras características interessantes que são inerentes a grafos minimais, como nesse caso o custo de implantaç ao da rede. A seguir, explicaremos o método e suas exceções.

Sendo G a árvore inicial com pelo menos 3 vértices, substituiremos todos os vértices de grau 2 ou maior por grafos desconexos D_k, sendo k igual a 2 vértices. Além disso, substituiremos todas as folhas por D_k, com k igual a 1 vértice.

Sejam D_k^i e D_k^j os substitutos dos vértices i e j, respectivamente. Se (i, j) são adjacentes em G, então cada vértice de D_k^i será unido a cada vértice de D_k^j. Se a árvore possuir 2 vértices, substituiremos todos os vértices por grafos desconexos D_2 formando o grafo gêmeo mais simples, o ciclo de 4 vértices. Percebe-se também que não há equivalentes para a árvore de 1 vértice, isto é, um vértice isolado.

A Figura 4 ilustra o mapeamento de árvore em um grafo gêmeo, obtido com a aplicação desse processo.

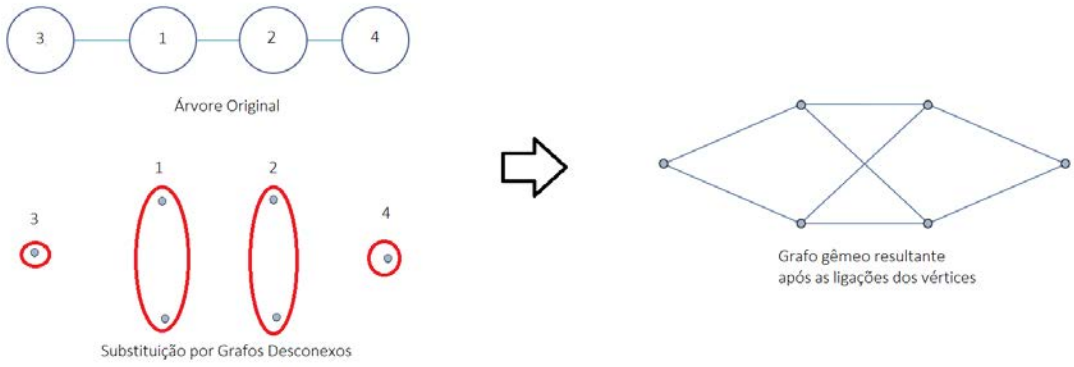

Figura 4: Exemplo de mapeamento de uma árvore em um grafo gêmeo.

4. Conclusão

Neste trabalho, foram investigados mapeamentos entre grafos gêmeos e árvores livres. Foi exibido um método para fazer uma decomposição em árvores a partir de um grupo específico de grafos, os grafos gêmeos, que possuem características interessantes para a modelagem de redes de telecomunicações. Esse método pode ser utilizado para o desenvolvimento de algoritmos eficientes para o tratamento desses grafos, como algoritmos de roteamento, por exemplo. Também foi exibido um método para a geração de grafos gêmeos a partir de árvores. Esse método pode ser utilizado para melhorar a eficiência dos algoritmos usados atualmente para gerar grafos gêmeos, favorecendo a ampliação dos estudos sobre essa classe de grafos.

Como continuação deste trabalho, serão implementadas rotinas computacionais para fazer mapeamentos de grafos gêmeos para árvores, e vice-versa. Para trabalhos futuros propõe-se uma melhor análise dos algoritmos beneficiados pela decomposição em árvores e a implementação desses em casos reais, utilizando os métodos propostos para quantificar a melhoria no tempo de execução dos algoritmos.

Agradecimentos

Este trabalho foi parcialmente financiado por FAPES, CAPES, and CNPq.

Referências

Chang, J.-M., Ho, C.-W., Hsu, C.-C., e Wang, Y.-L. (1996). The characterizations of hinge-free networks. In *Proc. Int. Comput. Symp. Algorithms*, p. 105–112, Taiwan.

Farley, A. M. e Proskurowski, A. (1997). Minimum self-repairing graphs. *Graphs and Combinatorics*, 13(4):345–351.

Nieminen, J. e Peltola, M. (1998). The minimum geodetic communication overlap graphs. *Applied Mathematics Letters*, 11(2):99 – 102.

OEIS (2010). OEIS, The On-Line Encyclopedia of Integer Sequences. Disponível em <http://oeis.org>. Visitado em 05/05/2016.

Paiva, M. H. M., Caporossi, G., e Segatto, M. E. V. (2013). Twin graphs for OTN physical topology design. *Les Cahiers du GERAD*, 48:1– 12.

Robertson, N. e Seymour, P. (1986). Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309 – 322.