



II CONPESQ Congresso de Pesquisa, Pós-Graduação e Inovação

Os novos rumos da ciência pós-pandemia

12 a 16 de abril de 2021 Universidade Federal do Cariri - UFCA

UMA DECOMPOSIÇÃO EM CAMINHOS DE GRADES

Maria Rayane Pereira de Oliveira¹

Centro de Ciência e Tecnologia,
Universidade Federal do Cariri
rayane.oliveira@aluno.ufca.edu.br

Thiago Braga Marcilon²

Centro de Ciência e Tecnologia,
Universidade Federal do Cariri
thiago.marcilon@ufca.edu.br

RESUMO: Neste trabalho é abordado o problema de decomposição em caminhos de grades. Grade é um tipo de grafo cujos vértices estão sob todas as coordenadas inteiras, (i, j) , e há uma aresta entre todos os vértices que estão a uma distância unitária. Um caminho simples é uma lista de vértices, sem repetição de arestas, de tal forma que, se u e v são vértices consecutivos na lista, (u, v) é uma aresta do grafo. Uma decomposição em caminhos de um grafo G é uma decomposição de G em caminhos simples de modo que cada aresta de G aparece exatamente em um dos caminhos. Nessa perspectiva, os principais resultados deste trabalho são os algoritmos polinomiais que retornam uma decomposição de tamanho mínimo em caminhos para grades e para um tipo especial de subgrafos de grades, ambos lineares no número de arestas do grafo de entrada. Os resultados obtidos são pioneiros e de grande relevância para a área de algoritmos em grafos, especialmente a de decomposição mínima em caminhos.

PALAVRAS-CHAVE: grafo; decomposição em caminhos; grade.

ABSTRACT: This paper addresses the problem of path decomposition of grids. Grid is a type of graph which its vertices lie in the integer coordinates (i, j) , and there is an edge between all vertices that are in a unit distance. A simple path is a list of vertices, without repeated edges, such that, if u and v are consecutive vertices in the list, then (u, v) is an edge of the graph. A path decomposition in a graph G is a decomposition of G in simple paths so that each edge of G appears in one of the paths. From this perspective, the main results of this paper are the algorithms that return a decomposition, of minimal length, into paths for grids and for a special type of grids subgraphs, both linear in the number of edges of the input graph. The results obtained are pioneer and of great relevance to the area of graph algorithms, specially of minimal path decomposition.

Keywords: graph; path decomposition; grid.

¹ Aluna do curso de Ciência da Computação. Bolsista UFCA no Programa Institucional de Iniciação Científica e Tecnológica da UFCA.

² Orientador

1 INTRODUÇÃO

Uma grade é um grafo cujos vértices estão sob todas as coordenadas inteiras (i, j) , tais que $1 \leq i \leq n$ e $1 \leq j \leq m$, e há uma aresta entre todos os vértices que estão a uma distância unitária. Define-se m como o número de colunas e n como o número de linhas da grade. Uma decomposição de um grafo G é um conjunto de subgrafos de G , onde cada aresta de G aparece em exatamente um dos subgrafos. Uma decomposição em caminhos é uma decomposição em que cada subgrafo é um caminho. A cardinalidade da menor decomposição em caminhos de G é $\rho n(G)$.

Os resultados na área de algoritmos de decomposição mínima em caminhos ainda são escassos, mas pode-se destacar resultados importantes como o de Constantinou e Ellinas (2018), no qual é mostrado um algoritmo polinomial que computa a menor decomposição em caminhos para alguns tipos de grafos bipartidos completos, e de Péroche (1984) que prova a **NP-Completeness** de alguns problemas de decomposição e cobertura em grafos. Botler e Cano (2019) apresentaram um modelo de programação inteira linear para calcular o tamanho da menor decomposição em caminhos de um grafo e é provada a Conjectura de Gallai para uma grande quantidade de grafos. No entanto, não existem resultados que abordam o problema de decomposição mínima em caminhos de grades.

Nesse sentido, as contribuições principais deste trabalho são os algoritmos que determinam uma menor decomposição em caminhos de grades e de alguns tipos de subgrafos de grades. Os algoritmos mostrados são lineares e possuem tempo de execução $O(|E|)$, onde E representa o conjunto de arestas do grafo de entrada G .

2 REFERENCIAL TEÓRICO

Em 1968, Tibor Gallai então concebeu a seguinte conjectura.

Conjectura de Gallai: *Todo grafo possui uma decomposição em caminhos de tamanho $\lceil n/2 \rceil$, onde n é o número de vértices do grafo.*

De fato, se a Conjectura de Gallai for verdadeira, temos então a resposta da pergunta feita por Erdős, já que conhecemos grafos cujo todas as decomposições por caminhos têm tamanho maior ou igual a $\lceil n/2 \rceil$, por exemplo, o grafo G de n vértices, que possui aresta entre

todos os pares de vértices, pois G possui $\frac{n(n-1)}{2}$ arestas e cada caminho subgrafo de G tem no máximo $n-1$ arestas e, portanto, toda decomposição em caminhos de G terá tamanho no mínimo $\lceil n/2 \rceil$.

Em 1968, Lovász (1968) dá um passo na direção da prova da Conjectura de Gallai e mostra que todo grafo G possui uma decomposição em caminhos e ciclos de tamanho menor ou igual a $\lceil n/2 \rceil$. Desde então, foram apresentados vários resultados provando a Conjectura de Gallai para várias classes específicas de grafos, porém, apesar da grande atenção dada à Conjectura de Gallai, até a presente data, ela permanece aberta no caso geral. De fato, existem muitos trabalhos relacionados a decomposição em caminhos do ponto de vista da Teoria dos Grafos, porém não existem muitos trabalhos que abordem esse assunto do ponto de vista algorítmico. De fato, até a presente data, os únicos trabalhos encontrados que possuem resultados algorítmicos relacionados a mínima decomposição em caminhos de grafos são:

- Péroche (1984) mostrou que o problema computacional determinar se existe uma decomposição em caminhos de tamanho dois é **NP-completo** mesmo para grafos de grau máximo 4;
- Constantinou e Ellinas (2018) apresentou um algoritmo polinomial para computar a menor decomposição em caminhos de alguns tipos de grafos bipartidos completos;
- Alspach (2008) apresentou caracterizações estruturais que implicam em um algoritmo polinomial para computar a menor decomposição em caminhos de grafos completos com um número par de vértices;
- Bryant (2010) apresentou caracterizações estruturais que implicam em um algoritmo polinomial para computar a menor decomposição em caminhos de grafos completos com um número ímpar de vértices.

3 RESULTADOS

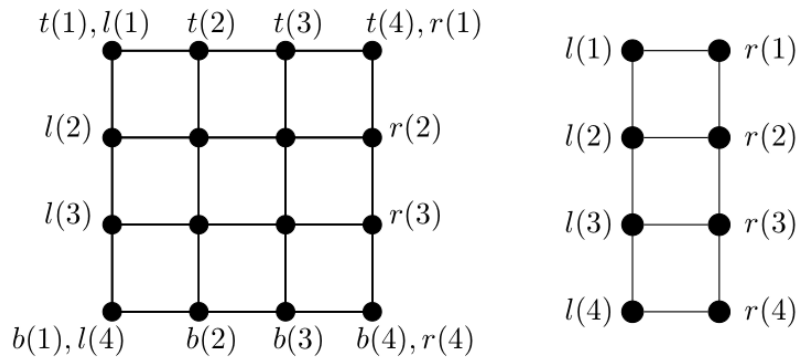
A partir da análise estrutural de grades e da busca por um algoritmo polinomial, foram sintetizados um algoritmo que, dados inteiros m e n , decompõe uma grade em caminhos; e um algoritmo que decompõe uma grade com arestas ausentes em uma das bordas de forma que cada uma das suas subgrades tem dimensões maiores do que 2.

3.1 DECOMPOSIÇÃO EM CAMINHOS DE GRADES

Os vértices e arestas de uma grade $G_{m \times n}$ são nomeados de acordo com as suas posições na grade e associando-os a um ou mais conjuntos entre direita, esquerda, superior e inferior.

Os vértices $r(i)$, $l(i)$, $t(j)$ e $b(j)$, para $i = (1, \dots, n)$ e $j = (1, \dots, m)$, são, respectivamente, aqueles que se encontram nas bordas direita, esquerda, superior e inferior, sendo R , L , T e B os conjuntos aos quais esses vértices pertencem. Considere como borda as extremidades da grade na qual todos os vértices têm grau diferente de 4. Note que os vértices de grau 2, definidos como vértices de canto, pertencem a dois conjuntos.

Figura 1: Exemplo das notações para $G_{4 \times 4}$ e $G_{2 \times 4}$



Fonte: Produção do próprio autor

Teorema 1: Determinar uma decomposição mínima de uma grade $G_{m \times n}$ pode ser feito em tempo linear.

Demonstração: Considere o Algoritmo 1. A seguir é mostrado que ele encontra uma decomposição mínima em tempo linear no número de arestas da grade. Considere, também, sem perda de generalidade, $m \leq n$.

Algoritmo 1: Decomposição em caminhos de grades

Entrada: inteiros m e n

Saída: uma decomposição em caminhos mínima de $G_{m \times n}$

- 1 se $(m \text{ ou } n) == 1$ então
- 2 P_1 é o maior caminho que contém todos os vértices
- 3 senão se $(m \text{ e } n) == 2$ então
- 4 $P_1 = \langle r(2), r(1), l(1) \rangle$
- 5 $P_2 = \langle l(1), l(2), r(2) \rangle$
- 6 senão se $m == 2$ então

7 P_1 é o caminho de $r(2)$ até $r(3)$ passando por $r(1)$, $l(1)$ e terminando com $l(3)$,
 $r(3)$
 8 $P_2 = \langle l(2), r(2), r(3) \rangle$
 9 **se** $n \geq 4$ **então**
 10 adicione $r(3)$, $r(4)$, $l(4)$, $l(3)$ a P_2
 11 **para** $i \leftarrow 3$ **até** $n - 2$
 12 $P_i = \langle r(i + 1), r(i + 2), l(i + 2), l(i + 1) \rangle$
 13 **senão**
 14 adicione $r(2)$, $t(m)$ a P_1
 15 **para** $k \leftarrow 1$ **até** $m - 2$ **faça**
 16 adicione $t(m - k + 1)$, $t(m - k)$ a P_k
 17 continue P_k percorrendo o menor caminho de $t(m - k)$ a $b(m - k)$
 18 adicione $b(m - k)$, $b(m - k - 1)$ a P_k
 19 adicione $b(1)$, $l(n - 1)$ a P_{m-2}
 20 adicione $t(2)$, $t(1)$ a P_{m-1}
 21 **para** $k \leftarrow 1$ **até** $n - 2$ **faça**
 22 adicione $l(k)$, $l(k + 1)$ a P_{k+m-2}
 23 continue P_{k+m-2} percorrendo o menor caminho de $l(k + 1)$ a $r(k + 1)$
 24 adicione $r(k + 1)$, $r(k + 2)$ a P_{k+m-2}
 25 adicione $r(n)$, $b(m - 1)$ a P_{m+n-4}
 26 **retorna** $\{P_i \mid 1 \leq i \leq m + n - 4\}$

Dados inteiros m e n , é mostrado como o algoritmo acima retorna de fato uma decomposição de $G_{m \times n}$ da seguinte maneira:

Quando uma das dimensões é 1, a grade é um caminho, o qual é retornado. No caso em que m e $n = 2$, já que a grade é um ciclo, o número mínimo de caminhos da menor grade é 2, os quais são retornados. Além disso, esses dois primeiros caminhos são explícitos e não há arestas repetidas. Se $m = 2$ e $n = 3$, os dois caminhos são bem definidos nas **linhas 7 e 8** e, como o grafo possui um ciclo, os dois caminhos compõem uma decomposição por caminhos mínima. Quando $m = 2$ e $n \geq 4$ os próximos caminhos são compostos pelas últimas 3 arestas, e logo, é uma decomposição.

Para os demais casos, note que existem $m P_n$ e $n P_m$, sendo P_m e P_n os caminhos de tamanho m e n , verticais e horizontais, respectivamente, que vão de uma extremidade à outra.

Na **linha 14** do algoritmo, a aresta $r(2)$, $t(m)$ é adicionada ao primeiro caminho

vertical. A partir do **para da linha 15**, uma última aresta é adicionada antes de o primeiro P_n ser percorrido. Feito isso, os próximos caminhos sempre começam de $t(i)$, e conseqüentemente, na execução da **linha 20** todos os P_n já foram percorridos, com exceção dos dois caminhos cujos vértices estão em R ou L . Para os caminhos horizontais, analogamente, adiciona-se uma última aresta antes de percorrer o primeiro P_m , e, como todos os caminhos começarão em $l(i)$, todos os P_m caminhos são percorridos, exceto os que estão em T ou B .

Além das arestas que estão em algum caminho, sobram as arestas das extremidades que são adicionadas exatamente ao iniciar ou terminar um caminho nas **linhas 16 e 23** ou nas **linhas 18 e 25**, e, por último, as arestas incidentes a vértices de grau par, que são adicionadas nas **linhas 20 e 27**. Perceba que, cada aresta adicionada a um caminho está bem definida e é única. Portanto, o algoritmo retorna uma decomposição de G .

Visto que, em qualquer decomposição em caminhos, cada vértice de grau ímpar certamente é extremidade de um dos caminhos, temos que $pn(G) \geq o/2$, onde o é o número de vértices de grau ímpar. Sendo assim, para o caso em que $n + m \geq 6$, o número de caminhos retornados na decomposição é sempre mínimo, pois sempre é retornada uma decomposição em caminhos de tamanho $m + n - 4 = o/2$.

■

3.2 DECOMPOSIÇÃO EM CAMINHOS DE ALGUNS TIPOS DE GRADES PARCIAIS

Nesta seção, será apresentado um algoritmo para determinar uma decomposição mínima em um tipo especial de subgrafos de grades. Os subgrafos de grades trabalhados são os subgrafos que consistem em uma grade com arestas ausentes em uma de suas bordas.

A ideia de abordar grades com arestas ausentes é útil, pois é um passo na direção de descobrir a complexidade e subgrafos induzidos de grade e grafos grade sólidos. Para tanto, segue-se a seguinte abordagem: cada aresta ausente separa a grade em subgrades completas, e, a partir delas, será possível fazer uma decomposição em caminhos em toda a grade.

Uma aresta de transição é aquela que está entre duas subgrades. O vértice oposto de um vértice v é o vértice que está no lado oposto, mas na mesma linha de v , denotado por v_o . A aresta oposta de uma aresta (u, v) é aquela com extremidades (u_o, v_o) . Por último, r é o total

Saída: uma decomposição em caminhos mínima de G

- 1 uma linha acima da primeira linha de G adicione uma subgrade com $n = 3$
- 2 uma linha abaixo da última linha de G adicione uma subgrade com $n = 3$
- 3 conecte as subgrades adicionadas com o restante do grafo, exceto na direita
- 4 decomponha as subgrades
- 5 una os caminhos através das arestas de transição
- 6 $k =$ número de caminhos até então
- 7 **para** $i \leftarrow 1$ **até** r **faça**
- 8 comece P_{k+i} da aresta (u, v) oposta à i -ésima aresta ausente
- 9 **se** $d(v_o) = 1$ **então**
- 10 continue percorrendo P_{k+i} seguindo o menor caminho de v a v_o
- 11 remova as subgrades e arestas adicionadas
- 12 **se** $d(t(m)) = 1$ **então**
- 13 **para** $i \leftarrow 1$ **até** $m - 1$ **faça**
- 14 adicione $t(m), t(m - 1)$ ao caminho que passa pela aresta de transição incidente a $t(m - 1)$
- 15 $k' =$ total de caminhos
- 16 **retorna** $\{P_i \mid 1 \leq i \leq k'\}$

Note que o fato de faltarem arestas em um lado da grade compromete o começo e o final de alguns caminhos, já que as arestas do meio da grade ainda existem. Algumas dessas arestas são chamadas de transição justamente porque conectam subgrades. As únicas arestas de transição que não unem caminhos são as que pertencem ao lado esquerdo e como suas extremidades são somente vértices de grau ímpar, é possível que se inicie um novo caminho nelas.

A união dos caminhos pelas arestas de transição será da seguinte forma: caso o caminho não tenha começado e terminado em uma só subgrade, a aresta de transição unirá o caminho que vem pela direita na subgrade superior com o caminho que vem pela esquerda na inferior.

A adição e remoção das novas subgrades $G_{m \times 3}$ e das arestas que vão ligá-las à grade original leva tempo polinomial, pois a quantidade de vértices e arestas que serão adicionadas é conhecida. Uma subgrade $G_{m \times 3}$ tem a quantidade de vértices igual a $3m$, e para saber a quantidade de arestas basta fazer o seguinte: uma grade $G_{m \times n}$ tem $n P_m$ e $m P_n$, como cada caminho P_k tem $k - 1$ arestas, logo, o número de arestas de $G_{m \times n}$ é

$m(n - 1) + n(m - 1)$. No caso em que $n = 3 \Rightarrow |E| = 3(m - 1) + 2m = 5m - 2$. Para unir as grades são necessárias $2(m - 1)$ arestas. Conclui-se, então, que são adicionados exatamente $6m$ vértices e $2(m - 1) + 10m - 4 = 12m - 8$ arestas.

A união de caminhos na **linha 5** pode ser feita, resumidamente, utilizando vetores. O primeiro tipo de vetor armazena, para cada vértice ímpar nas subgrades, o caminho que inicia e termina nele; o segundo armazena, para cada caminho, os vértices que estão nas suas extremidades. Cada vez que um caminho P_i , que vem pela aresta de transição da subgrade superior, chega na subgrade de baixo os valores dos vetores podem ser mudados da seguinte maneira: o vértice que antes começava um caminho P_j na subgrade de baixo passa a ser um vértice intermediário de P_i , e basta ver qual é a extremidade final de P_j e torná-la extremidade final de P_i . Assim, as informações de P_j e dos vértices que eram suas extremidades são deletadas. As operações nos vetores levam $O(1)$ para $O(|E|)$ caminhos e $O(n)$ vértices, totalizando $O(|E| + n) = O(|E|)$.

Sabendo disso, pode-se afirmar que o algoritmo é polinomial, pois a **linha 4** leva $O(|E|)$ para cada subgrade; a **linha 5** é a operação de unir caminhos; e, para o caso de entrar na **linha 13** são realizadas $m - 1$ alterações em caminhos já existentes. Portanto, o algoritmo executa em $O(|E|)$.

Ademais, o algoritmo retorna uma decomposição em caminhos porque cada subgrade é decomposta seguindo o Algoritmo 1 e o **para** da **linha 7** decompõe as arestas que não estão em subgrades.

Todos os caminhos percorridos na **linha 4** começam e terminam em um vértice de grau ímpar se considerarmos cada subgrade individualmente, ou seja, que elas não estão ligadas por arestas de transição. No entanto, cada aresta de transição confere a suas extremidades um grau par, o que permite que esses caminhos possam ser continuados. Devido à forma que a união é feita, unindo o caminho que vem pela direita na subgrade superior com o caminho que vem pela esquerda na inferior, apenas a quantidade de caminhos é reduzida, não alterando a paridade de suas extremidades. Note que, quando as subgrades adicionadas são removidas essa propriedade se mantém, já que as arestas de transição que as ligava ao restante da grade também são removidas.

Na saída do algoritmo, todos os vértices ímpares terão começado e/ou terminado exatamente um caminho, já que no Algoritmo 1 é mostrado quais são os vértices que iniciam

e terminam cada um dos caminhos.

■

4 CONCLUSÃO

Neste trabalho, foi estudado o problema de determinar uma decomposição mínima em caminhos de grades. As principais contribuições deste trabalho foram os algoritmos que retornam uma decomposição mínima para grades e para grades com arestas ausentes em uma das bordas, ambos sendo lineares no número de arestas da grade. Esses resultados são importantes dada a ausência de resultados algorítmicos no que diz respeito à decomposição de grafos em caminhos.

Por fim, alguns problemas relacionados aos tratados aqui continuam em aberto, como o problema de decomposição em caminhos para subgrafos induzidos de grades, para grafos grade sólidos e em grades parciais.

AGRADECIMENTOS

Agradecimentos à Universidade Federal do Cariri pelo financiamento do presente trabalho.

REFERÊNCIAS

ALSPACH, Brian. The wonderful walecki construction. **Bulletin of the Institute of Combinatorics and its Applications**, v. 52, p. 7 – 20, 2008.

BONAMI, Marthe; PERRETT, Thomas J. Gallai's path decomposition conjecture for graphs of small maximum degree. **Discrete Mathematics**, v. 342, n. 5, p. 1293-1299, 2019. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0012365X19300081> Acesso em 05 fev. 2021.

BOTLER, Fábio; CANO, R.; SAMBINELLI, M. On Computing the Path Number of a Graph. **Electronic Notes in Theoretical Computer Science**, v. 346, p. 185-197, 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1571066119300672> Acesso em 05 fev. 2021.

BOTLER, Fábio; JIMÉNEZ, Andrea. On path decompositions of $2k$ -regular graphs. **Discrete Mathematics**, v. 340, n. 6, p. 1405-1411, 2017. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0012365X1630317X> Acesso em 05 fev. 2021.

BRYANT, Darryn. Packing paths in complete graphs. **Journal of Combinatorial Theory Series B**, v. 100, p. 206 – 215, 2010. Disponível em: <https://www.sciencedirect.com/science/article/pii/S009589560900063X>. Acesso em 12 fev. 2021.

CONSTANTINOU, Costas K.; ELLINAS, Georgios. Minimal Path Decomposition of Complete Bipartite Graphs. **Journal of Combinatorial Optimization**, v. 35, n. 3, p. 684-702, 2018. Disponível em: https://www.researchgate.net/publication/321074170_Minimal_path_decomposition_of_complete_bipartite_graphs Acesso em: 05 fev. 2021.

JIMÉNEZ, Andrea; WAKABAYASHI, Yoshiko. On path-cycle decompositions of triangle-free graphs. **Discrete Mathematics and Theoretical Computer Science**, v. 19, 2017. Disponível em: https://www.researchgate.net/publication/260231521_On_path-cycle_decompositions_of_triangle-free_graphs Acesso em 05 fev. 2021.

LOVÁSZ, László. On covering of graphs. **Theory of Graphs**. 1968. Disponível em: https://www.researchgate.net/publication/268652766_On_covering_of_graphs Acesso em 05 fev. 2021.

PÉROCHE, B. NP-completeness of some problems of partitioning and covering in graphs. **Discrete Applied Mathematics**, v. 8, n. 2, p. 195-208, 1984. Disponível em: <https://www.sciencedirect.com/science/article/pii/0166218X8490101X> Acesso em 05 fev. 2021.