

Emparelhamentos Conexos

Bruno P. Masquio¹, Paulo E. D. Pinto^{1*}, Jayme L. Szwarcfiter^{1,2*}

¹Universidade do Estado do Rio de Janeiro, Brasil

²Universidade Federal do Rio de Janeiro, Brasil

brunomasquio@ime.uerj.br, pauloedp@ime.uerj.br, jayme@nce.ufrj.br

Abstract. *Graph matching problems are well known and studied, in which we want to find sets of pairwise non-adjacent edges. Recently, there has been an interest in the study of matchings in which the induced subgraphs by the vertices of matchings are connected or disconnected. Although these problems are related to connectivity, the two problems are probably quite different, regarding their complexity. While the complexity of finding a maximum disconnected matching is still unknown for a general graph, the one for connected matchings can be solved in polynomial time. Our contribution in this paper is a linear time algorithm to find a maximum connected matching of a general connected graph, given a general maximum matching as input.*

1. Introdução

Emparelhamentos em grafos é uma área muito estudada e com muitas aplicações. Recentemente, surgiu o interesse de estudar emparelhamentos restritos a subgrafos [Masquio 2019] [Goddard et al. 2005]. Nessa abordagem, procura-se resolver o problema tal que certas propriedades de subgrafos induzidos pelos vértices do emparelhamento sejam satisfeitas. Algumas dessas propriedades, por exemplo, são que esses subgrafos sejam 1-regulares, acíclicos, conexos ou desconexos.

Inicialmente, apresentaremos algumas notações utilizadas no artigo. Considere G um grafo, n o número de vértices, m o número de arestas e M um emparelhamento de G . Denotamos por $G[M]$ o subgrafo de G induzido pelos vértices incidentes nas arestas de M . Denotamos por $N(v)$ o conjunto de vértices vizinhos de v em G . Se existe uma aresta de M incidente a v , então v é M -saturado. Além disso, usamos $\beta(G)$ para a cardinalidade de um emparelhamento máximo. Um subconjunto $S \subset V(G)$ é dito um *separador* de G se existem dois vértices que estão na mesma componente conexa de G e em componentes conexas distintas de $G - S$. Também dizemos que S é um *separador minimal* de G se S é um separador e não há subconjunto próprio de S que também seja um separador.

Um emparelhamento M é dito *conexo* se $G[M]$ é conexo. Usamos $M_c(G)$ para denotar um emparelhamento conexo máximo e $\beta_c(G)$ para sua cardinalidade. Analogamente, um emparelhamento M é *desconexo* se $G[M]$ é desconexo. Denotamos como $M_d(G)$ um emparelhamento desconexo máximo e $\beta_d(G)$ a sua cardinalidade. A Figura 1 mostra dois emparelhamentos máximos em um grafo. O primeiro, na figura da esquerda, é um emparelhamento desconexo máximo e o segundo, na figura da direita, um emparelhamento conexo, também máximo.

*Projeto parcialmente financiado por FAPERJ.

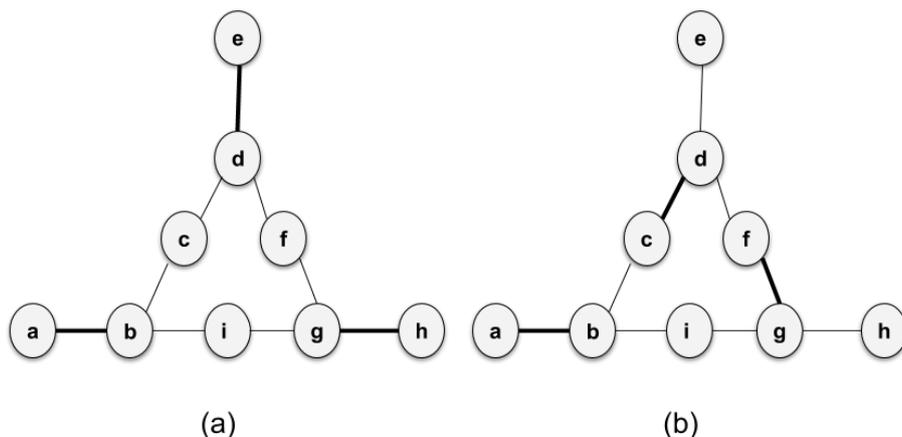


Figura 1. Exemplo de emparelhamentos desconexo e conexo

Neste artigo discutimos a solução para emparelhamentos conexos e a contrastamos com resultados parciais existentes para emparelhamentos desconexos. O principal resultado mostrado é um algoritmo de complexidade linear para obter um emparelhamento conexo máximo, dado um emparelhamento máximo qualquer do grafo.

2. Emparelhamento conexos para grafos em geral

O problema de emparelhamento conexo máximo para grafos em geral pode ser resolvido a partir do seguinte teorema.

Teorema 1. *Se G é conexo, então $\beta_c(G) = \beta(G)$ [Goddard et al. 2005].*

Embora os autores do Teorema 1 tenham apresentado uma solução para o problema, eles não apresentaram sua complexidade nem um algoritmo para obter o emparelhamento conexo máximo. Nossa contribuição é um algoritmo linear para esse propósito, usando como entrada um emparelhamento máximo, que pode ser desconexo.

A ideia do algoritmo, baseada no Teorema 1, é que, dado um emparelhamento máximo M , é possível construir um emparelhamento conexo de mesma cardinalidade apenas alterando arestas de M , sem alterar a sua cardinalidade.

Vamos descrever como a alteração de arestas pode ser feita. Seja um emparelhamento máximo M tal que $G[M]$ seja desconexo e r um vértice qualquer M -saturado. Considere C_r a componente de $G[M]$ que contém r e $W_r = \{w | w \in N(v) \setminus V(C_r), \forall v \in C_r\}$, ou seja, W_r é o conjunto de vértices fora da componente C_r que são vizinhos de algum vértice dessa componente. Podemos observar que, em G , existe um vértice saturado $u \notin C_r$ adjacente a algum vértice w de W_r . Logo, é possível aumentar a cardinalidade de C_r removendo de M a aresta saturada de u e adicionando a aresta (u, w) . Esse procedimento pode ser feito até que $G[M]$ seja conexo.

O algoritmo utiliza os conjuntos Q_s e Q_n , que guardam vértices saturados e não saturados por M , respectivamente, C , no qual são inseridos vértices de C_r ou novos vértices adicionados a essa componente e W , que guarda vizinhos de vértices de C_r fora da componente. O laço(loop) principal adiciona pelo menos um vértice a C , a cada iteração, e contém dois outros laços auxiliares. O primeiro laço auxiliar, executa EXPANDE(v) para cada vértice v de Q_s , analisa $N(v)$ e adiciona adequadamente cada vértice dessa

vizinhança ainda não adicionado. Já o segundo laço auxiliar, executa a função TENTACONECTAR(v) para cada vértice v de Q_n . Caso exista $w \in N(v) \setminus C$, então w está saturado por uma aresta (w, u) e é feita a operação de troca de arestas em M , removendo a aresta (w, u) e adicionando a aresta (v, w) .

Algoritmo 1 Emparelhamento conexo máximo

```

1: função EXPANDE(Vértice:  $v$ )
2:   para  $w \in N(v) \setminus C$  faça
3:     se  $w$  está  $M$ -saturado então
4:        $C \leftarrow C \cup \{w\}$ ;  $Q_s \leftarrow Q_s \cup \{w\}$ 
5:     senão se  $w \notin W$  então
6:        $W \leftarrow W \cup \{w\}$ ;  $Q_n \leftarrow Q_n \cup \{w\}$ 
7:
8: função TENTACONECTAR(Vértice:  $v$ )
9:   se  $\exists w \in N(v) \setminus C$  então
10:     $u \leftarrow$  Vértice emparelhado com  $w$  em  $M$ 
11:     $M \leftarrow M \setminus \{(u, w)\} \cup \{(v, w)\}$ 
12:     $C \leftarrow C \cup \{v, w\}$ 
13:     $W \leftarrow W \setminus \{v\}$ 
14:     $Q_s \leftarrow Q_s \cup \{v, w\}$ 
15:
16: #Entrada:  $G$ : Grafo,  $M$ : emparelhamento máximo em  $G$ 
17:  $r \leftarrow$  Vértice qualquer  $M$ -saturado
18:  $C \leftarrow \{r\}$ ;  $Q_s \leftarrow \{r\}$ ;  $W \leftarrow \emptyset$ ;  $Q_n \leftarrow \emptyset$ ;
19: enquanto  $|C| < |M| * 2$  faça
20:   enquanto  $|Q_s| > 0$  faça
21:      $v \leftarrow$  Vértice qualquer de  $Q_s$ 
22:      $Q_s \leftarrow Q_s \setminus \{v\}$ 
23:     EXPANDE( $v$ )
24:   enquanto  $|Q_n| > 0$  faça
25:      $v \leftarrow$  Vértice qualquer de  $Q_n$ 
26:      $Q_n \leftarrow Q_n \setminus \{v\}$ 
27:     TENTACONECTAR( $v$ )
28: retorna  $M$ 

```

Teorema 2. O Algoritmo 1 tem complexidade de tempo $O(n + m)$.

Demonstração. Observe que, em princípio, $G[M]$ não é necessariamente conexo. O laço principal é executado enquanto não certificamos que $G[M]$ é conexo. Tal confirmação é feita somente quando $|C| = |M| * 2$. Os vértices entram em Q_s uma única vez e seus vizinhos são processados na função EXPANDE(v), enquanto aqueles que entram em Q_n , no máximo uma vez, têm seus vizinhos processados em TENTACONECTAR(v), sendo que eles podem migrar para Q_s . Portanto, a vizinhança de cada vértice é examinada no máximo duas vezes, o que leva à complexidade $O(n + m)$.

□

Como exemplo do uso do Algoritmo 1, se tivéssemos o grafo da Figura 1 e o emparelhamento inicial máximo (e desconexo) fosse o da esquerda da figura, $M = \{(a, b), (d, e), (g, h)\}$, esse emparelhamento poderia ser sucessivamente modificado para $M = \{(a, b), (c, d), (g, h)\}$ e $M = \{(a, b), (c, d), (f, g)\}$. Este último é um emparelhamento conexo e está mostrado na parte direita da figura.

Consideremos agora a complexidade de obtenção de um emparelhamento máximo M , que é entrada para o algoritmo descrito. Sabemos que um emparelhamento máximo pode ser obtido em $O(m\sqrt{n})$ [Micali and Vazirani 1980] para grafos em geral. Logo, essa complexidade também vale para um emparelhamento conexo máximo, ou seja, o problema tem solução polinomial. Além disso, são conhecidas algumas classes, como árvores e grafos bloco, cujos emparelhamentos máximos podem ser obtidos em tempo linear[Masquio 2019]. Para essas classes, considerando o Algoritmo 1, o problema do emparelhamento conexo máximo tem complexidade linear.

3. Um contraste entre emparelhamentos desconexos e conexos

O problema de encontrar um emparelhamento desconexo máximo em um grafo qualquer está em aberto. Uma idéia de solução é a de retirar do grafo, alternativamente, cada um dos seus separadores minimais S , determinar o emparelhamento máximo M de $G - S$ que maximiza a cardinalidade de M , desde que M seja desconexo. Entretanto, essa abordagem leva a um algoritmo que pode ser exponencial, pois o grafo pode ter um número exponencial de separadores minimais. Devido a essa dificuldade, conjecturamos que o problema seja NP-completo. Apesar disso, para algumas classes, o problema pode ser resolvido em tempo polinomial como, por exemplo, caminhos, ciclos, árvores e grafos bloco, para os quais foram encontrados algoritmos lineares[Masquio et al. 2019]. Nesse mesmo trabalho foi mostrado um algoritmo de complexidade $O(nm)$ para grafos cordais. Mencionamos ainda que não se conhece solução especial para grafos completos ou split.

Em contraste, o problema de determinar emparelhamentos conexos máximos tem solução polinomial para grafos em geral. Neste trabalho mostramos ainda que, se for dado um emparelhamento máximo do grafo, o emparelhamento conexo máximo pode ser obtido com complexidade linear.

Referências

- Goddard, W., Hedetniemi, S. M., Hedetniemi, S. T., and Laskar, R. (2005). Generalized subgraph-restricted matchings in graphs. *Discrete Math.*, 293(1-3):129–138.
- Masquio, B. P. (2019). Emparelhamentos desconexos. Master's thesis, Universidade do Estado do Rio de Janeiro. Disponível em <https://github.com/BMasquio/papers/raw/master/MastersThesisBrunoMasquio.pdf>.
- Masquio, B. P., Pinto, P. E. D., and Szwarcfiter, J. L. (2019). Algoritmos Eficientes Para Emparelhamentos Desconexos em Grafos Cordais e Grafos Bloco. In *Anais do Encontro de Teoria da Computação (ETC/CSBC)*, Natal - RN. SBC.
- Micali, S. and Vazirani, V. V. (1980). An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs. In *21st Ann. Symp. on Foundations of Comp. Sc.*, pages 17–27.