

# O produto forte de um grafo não trivial e o grafo completo possui 2- e 3-atribuição de papéis

Gustavo Morais Medeiros<sup>1</sup>, Julliano Rosa Nascimento<sup>1</sup>

<sup>1</sup>Instituto de Informática – Universidade Federal de Goiás (UFG)  
Caixa Postal 131 – 74001-970 – Goiânia – GO – Brasil

medeiros@discente.ufg.br, jullianonascimento@ufg.br

**Abstract.** *Let  $G$  be a simple graph and  $r$  a positive integer. With applications in social network analysis, an  $r$ -role assignment is an assignment of  $r$  distinct roles to the vertices of  $G$ , such that two vertices with the same role share the same set of roles among their adjacent vertices. It is known that the problem of deciding whether an arbitrary graph  $G$  has an  $r$ -role assignment is  $\mathcal{NP}$ -complete for  $r \geq 2$ , and there are few results in the literature on this problem for graph products. In this regard, we present a linear-time algorithm to determine 2- and 3-role assignments for the strong product of a nontrivial graph  $G$  and  $K_n$ ,  $n \geq 2$ , which we conclude that the strong product of  $G$  and  $K_n$  always has 2- and 3-role assignments.*

**Resumo.** *Seja  $G$  um grafo simples e  $r$  um inteiro positivo. Com aplicações em análise de redes sociais, uma  $r$ -atribuição de papéis é uma atribuição de  $r$  papéis distintos aos vértices de  $G$ , tal que, dois vértices com o mesmo papel têm o mesmo conjunto de papéis nos vértices adjacentes. Sabe-se que o problema de se decidir se um grafo arbitrário  $G$  possui uma  $r$ -atribuição é  $\mathcal{NP}$ -completo para  $r \geq 2$  e há poucos resultados na literatura sobre esse problema para produtos de grafos. Nesse aspecto, apresentamos um algoritmo linear para determinar 2- e 3-atribuição de papéis para o produto forte entre um grafo não trivial  $G$  e  $K_n$ ,  $n \geq 2$ , do qual concluímos que o produto forte entre  $G$  e  $K_n$  sempre possui 2- e 3-atribuição de papéis.*

## 1. Introdução

A Teoria dos Grafos é um tema bastante estudado e importante, pelo fato de que vários problemas em muitas áreas podem ser resolvidos usando modelos de grafos. Essa é uma área da matemática que teve sua origem no século XVIII, e desde o seu desenvolvimento, muitas descobertas foram feitas a partir da definição básica de grafo. Um grafo  $G = (V, E)$  consiste de um conjunto  $V$  não vazio de vértices e um conjunto  $E$  de pares não-ordenados de vértices, chamados arestas. Conceitos básicos sobre grafos podem ser encontrados em [1, 5].

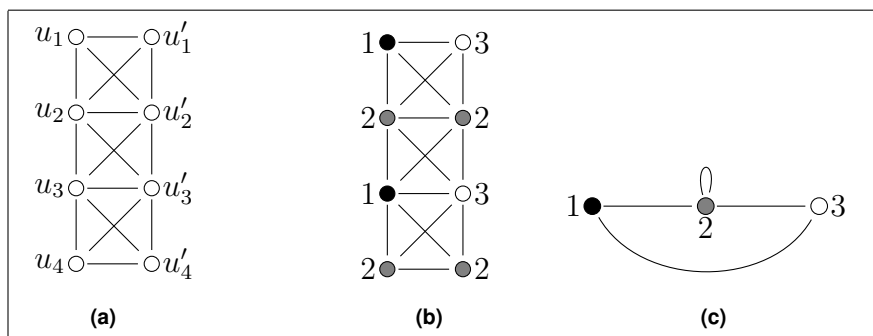
Um tema bastante estudado em grafos é a coloração (própria) de vértices [10], que seria uma forma de atribuir cores aos vértices de um grafo  $G$ , de forma que não hajam vértices adjacentes com a mesma cor.

O foco deste artigo se concentra na atribuição de papéis [7], uma variação da coloração. Na atribuição de papéis, passamos a considerar que um vértice pode ter elemento adjacente com a mesma cor. Porém, as cores dos vizinhos de um vértice que

recebe uma determinada atribuição devem ser iguais para todos os vértices que recebem essa mesma atribuição.

A atribuição de papéis é uma função de homomorfismo [11], que intuitivamente, conserva a estrutura de um grafo, podendo reduzi-lo e manter algumas de suas propriedades. Esse ponto de vista é importante pela facilidade que pode proporcionar aos estudos e trabalhos com grandes grafos. Por exemplo, para extração de informações em grafos de interações e papéis desempenhados por indivíduos em uma rede social. Sejam  $G$  e  $R$  dois grafos, uma  $R$ -atribuição de papéis para  $G$  é um mapeamento de vértices  $p : V(G) \rightarrow V(R)$ , de modo que a relação de vizinhança é mantida, ou seja, todos os papéis vizinhos à imagem de um vértice aparecem como papéis na vizinhança do vértice. Tal condição pode ser formalmente expressa como: para todo  $u \in V(G) : p(N_G(u)) = N_R(p(u))$  [12]. Se  $|V(R)| = r$ , então temos uma  $r$ -atribuição de papéis. Um exemplo de 3-atribuição de papéis segue na Figura 1(b).

Existem algumas operações que podem ser aplicadas a grafos, o produto é uma delas [9]. Há resultados de atribuição de papéis para alguns produtos de grafos, por exemplo: Cartesiano [3, 12, 18], direto [8], lexicográfico [18] e prisma complementar [2, 12]. Neste artigo, focaremos no produto forte de grafos, que, até onde sabemos, não possui resultados em relação à atribuição de papéis. Dados dois grafos  $G$  e  $H$ , o *produto forte* de  $G$  e  $H$  é denotado por  $G \boxtimes H$  e é definido como segue: um grafo com conjunto de vértices  $V(G \boxtimes H) = \{(g, h) \mid g \in V(G) \text{ e } h \in V(H)\}$  e o conjunto de arestas  $E(G \boxtimes H) = \{(g, h)(g', h') \mid g = g', hh' \in E(H) \text{ ou } gg' \in E(G), h = h'\} \cup \{(g, h)(g', h') \mid gg' \in E(G) \text{ e } hh' \in E(H)\}$  [9]. Notamos que  $E(G \boxtimes H)$  é obtido pela união do conjunto de arestas do produto Cartesiano e produto direto dos grafos  $G$  e  $H$ . Um exemplo de produto forte  $P_4 \boxtimes K_2$  segue na Figura 1(a).



**Figura 1. (a) Grafo  $P_4 \boxtimes K_2$ .  
 (b) Uma 3-atribuição para o grafo  $P_4 \boxtimes K_2$  com o grafo de papéis  $R_2$ .  
 (c) Grafo  $R_2$ .**

No que diz respeito à complexidade computacional da atribuição de papéis, em 2001, Roberts e Sheng [16] demonstraram que o problema de decisão associado a 2-atribuição de papéis é um problema  $\mathcal{NP}$ -completo para grafos gerais. Em 2005, Fiala e Paulusma [8] estenderam o resultado, mostrando que  $r$ -atribuição de papéis é  $\mathcal{NP}$ -completo para  $r \geq 3$  fixo.

Em uma abordagem mais específica, Van't Hof, Paulusma e Van Rooij [17] estabeleceram uma dicotomia para a complexidade da  $r$ -atribuição de papéis em grafos cordais. Permanece  $\mathcal{NP}$ -completo para  $r \geq 3$ , mas é solucionável em tempo linear para  $r = 2$ . Para grafos planares, Purcell e Rombach [14] provaram que a  $r$ -atribuição de papéis, com

$r \geq 2$ , permanece  $\mathcal{NP}$ -completo. No entanto, para cografos, a complexidade do problema é constante.

Dourado [6] determinou uma dicotomia para grafos *split* na complexidade da  $r$ -atribuição de papéis. É trivial para  $r = 2$ , solucionável em tempo linear para  $r = 3$  e  $\mathcal{NP}$ -completo para  $r \geq 4$ .

Em 2019, Pandey [13] mostrou que a  $r$ -atribuição de papéis, com  $r \geq 3$ , é  $\mathcal{NP}$ -completo para grafos bipartidos conexos. Para  $r = 2$ , a complexidade é constante.

Para grafos com grau máximo 3, Purcell e Rombach [15] provaram que o problema de  $r$ -atribuição de papéis, com  $r \geq 2$ , continua  $\mathcal{NP}$ -completo.

Para os grafos livres de união disjunta de duas cópias de  $K_2$ , Pandey [13] mostrou que  $r$ -atribuição de papéis é trivial para  $r = 3$  e  $\mathcal{NP}$ -completo para  $r \geq 4$ , com base na pesquisa de Dourado [6].

Mesquita [4], provou que para produto Cartesiano sempre existe uma 2-atribuição de papéis, e ainda assim o problema permanece  $\mathcal{NP}$ -completo para qualquer  $r \geq 3$  fixo. É mostrado também que 3-atribuição de papéis para prismas complementares pode ser resolvido em tempo linear. E por último, Mesquita conjectura que, para  $r \geq 3$ ,  $(r + 1)$ -atribuição de papéis para prismas complementares é  $\mathcal{NP}$ -completo.

Este estudo propõe algoritmos para a determinação de 2- e 3-atribuição de papéis para produto forte de um grafo arbitrário  $G$  com o grafo completo  $K_n$ , para  $n \geq 2$ . Além disso, será estabelecida a complexidade computacional desses algoritmos, demonstrando que ambos são algoritmos lineares. Em decorrência dos algoritmos apresentados, concluímos que  $G \boxtimes K_n$  sempre possui 2- e 3-atribuição de papéis.

Este artigo está estruturado como segue. Conceitos e notações básicas são apresentados na Seção 2. Nossos resultados seguem na Seção 3 e considerações finais seguem na Seção 4.

## 2. Conceitos Básicos

Um *grafo*  $G$  é um par ordenado  $(V(G), E(G))$  que consiste de um conjunto  $V(G)$  de *vértices* e um conjunto  $E(G)$  de *arestas*, disjunto de  $V(G)$ , de forma que  $E(G) \subseteq [V(G)]^2$ , ou seja, um elemento de  $E(G)$  é subconjunto de dois elementos de  $V(G)$ . Um *grafo simples* é um grafo que não permite laços e arestas paralelas. A quantidade de vértices de um grafo  $G$  é dito ser a *ordem* de  $G$ , em geral denotado por  $n$ . Um grafo de ordem 1 é dito *trivial* e de ordem pelo menos 2 é dito *não trivial*.

Denotamos uma aresta entre os vértices  $u$  e  $v$  por  $uv$ . Se  $uv \in E(G)$ , dizemos que o vértice  $u$  é *adjacente* ao vértice  $v$ , ou que  $u$  é *vizinho* de  $v$ . Para vizinhança de um vértice  $v$  usamos a notação  $N_G(v)$ . Denotamos ainda  $d(v)$  o *grau* de um vértice, que é o número de arestas incidentes a  $v$ .

Um grafo *completo*  $K_n$  é um grafo em que o grau de todos os vértices é igual a  $n - 1$ , ou seja, todos os vértices possuem adjacência entre si. Um vértice é dito *universal* quando  $N_G(v) = V(G) \setminus \{v\}$ .

Utilizaremos uma notação semelhante à usada em produtos Cartesianos por Mesquita [4]. Em um produto de grafo  $G \boxtimes K_2$  temos que  $K_2$  é um grafo completo de ordem

2, consideramos  $V(K_2) = \{v_1, v_2\}$ , para cada  $u \in V(G)$ , denotamos por  $u$  o vértice  $(u, v_1)$  e por  $u'$  o vértice  $(u, v_2)$  para qualquer  $u \in V(G)$ . Dizemos que  $u'$  é o vértice *correspondente* de  $u$ . Veja um exemplo na Figura 1(a).

Seja  $G$  um grafo simples e  $R$  um grafo possivelmente com laço. Um *homomorfismo* de grafo de  $G$  para  $R$  é uma função  $p : V(G) \rightarrow V(R)$  tal que  $p(u)p(v) \in E(R)$  sempre que  $uv \in E(G)$ . Se um homomorfismo  $p$  é *sobrejetor* e a restrição de  $p$  à vizinhança de cada  $u \in V(G)$  é sobrejetora, isto é, a função  $p_u : N_G(u) \rightarrow N_R(p(u))$ , então  $p$  é dito um *homomorfismo localmente sobrejetor* de  $G$  para  $R$  ou uma  *$R$ -atribuição de papéis* de  $G$ .  $R$  é chamado *grafo de papéis*. Para um inteiro  $r \geq 1$ , uma  *$r$ -atribuição de papéis* em um grafo  $G$  é uma função sobrejetora  $p : V(G) \rightarrow \{1, \dots, r\}$  onde  $\{p(u') : u' \text{ em } N(u)\} = \{p(v') : v' \text{ em } N(v)\}$  para cada par  $u, v$  em  $V(G)$  com  $p(u) = p(v)$ . Os elementos de  $\{1, \dots, r\}$  são chamados de *papéis*, ou *cores*. Exemplificando essa atribuição, o grafo da Figura 1(c) é o grafo de papéis do grafo da Figura 1(b).

Considere  $V(G) = \{u_1, u_2, \dots, u_m\}$  e  $V(H) = \{v_1, v_2, \dots, v_n\}$ . Chamamos de *linha*  $\mathcal{L}_i$  o conjunto de vértices  $\{(u_i, v_1), (u_i, v_2), \dots, (u_i, v_n)\}$  de  $V(G \boxtimes H)$ , e de *coluna*  $\mathcal{C}_j$  o conjunto de vértices  $\{(u_1, v_j), (u_2, v_j), \dots, (u_m, v_j)\}$  de  $V(G \boxtimes H)$ . Dada uma linha  $\mathcal{L}_i$  com o conjunto de vértices  $\{(u_i, v_1), (u_i, v_2), \dots, (u_i, v_n)\}$  de  $V(G \boxtimes H)$  definimos uma *linha adjacente* a  $\mathcal{L}_i$  a linha  $\mathcal{L}_{i+1}$  com o conjunto de vértices  $\{(u_{i+1}, v_1), (u_{i+1}, v_2), \dots, (u_{i+1}, v_n)\}$  de  $V(G \boxtimes H)$ .

Para 2- e 3-atribuição de papéis temos alguns possíveis grafos  $R$ . Nossos resultados usam explicitamente  $R_1$  e  $R_2$  representados nas Figuras 2 e 1(c), respectivamente.

Definimos um *loop* como um conjunto de instruções que se repete em um algoritmo, continuando por um número determinado de iterações ou até que uma condição específica encerre o processo.

### 3. Resultados

Nesta seção apresentamos um algoritmo de rotulação para um grafo  $G$  com algumas propriedades utilizadas para definir posteriormente funções para 2- e 3-atribuição de papéis para  $G \boxtimes K_2$ . Em seguida, aproveitamos as funções definidas para obtermos funções de 2- e 3-atribuição de papéis para  $G \boxtimes K_n$ . Por fim, analisamos a complexidade de execução do algoritmo e das funções.

#### 3.1. Algoritmo de rotulação para $G$

O Algoritmo 1 a seguir tem a finalidade de rotular os vértices do grafo  $G$  por meio da função  $\ell : V(G) \rightarrow \{1, 2\}$ , função esta que será utilizada para auxiliar a definição de funções de atribuição mais adiante.

O objetivo do Algoritmo 1 consiste em associar rótulos aos vértices através da definição da função  $\ell$ , de tal forma a manter a propriedade de que um vértice de rótulo 1 sempre possua vizinhos de rótulo 2 e não possua vizinhos de rótulo 1, e um vértice de rótulo 2 sempre possua vizinhos de rótulo 1.

Para realizar essa tarefa, é inicializado um rótulo com 0 para cada vértice. A partir de um vértice removido de uma fila  $Q$ , é atribuído o rótulo 1 se ele possui o rótulo 0, e toda a sua vizinhança é marcada com o rótulo 2, sendo esses adicionados à fila  $Q$ . Na sequência, a função binária  $a(v)$ , inicialmente definida como 0 para indicar que o vértice

não foi visitado, é alterada para o valor 1. Isso é feito com o intuito de manter o controle dos vértices já visitados e evitar que eles sejam novamente enfileirados. Caso o vizinho de um vértice removido tenha rótulo 2, todos os seus vizinhos são enfileirados e marcados como visitados.

Todo o processo descrito no parágrafo anterior é repetido até que a fila  $Q$  esteja vazia.

---

**Algoritmo 1:** Definição de  $\ell$ .

---

**Entrada:** Grafo  $G$ .  
**Saída:** Rotulação  $\ell : V(G) \rightarrow \{1, 2\}$ .

- 1 **Inicialização:**
- 2     Cria fila  $Q$  vazia;
- 3     **Para todo**  $v \in V(G)$  **faça**
- 4          $\ell(v) \leftarrow 0$ ;
- 5          $a(v) \leftarrow 0$
- 6 escolha  $v \in V(G)$  arbitrário;
- 7 enfileira( $v, Q$ );
- 8  $a(v) \leftarrow 1$ ;
- 9 **Enquanto**  $Q \neq \emptyset$  **faça**
- 10      $u \leftarrow$  desenfileira( $u, Q$ );
- 11     **Se**  $\ell(u) = 0$  **então**
- 12          $\ell(u) \leftarrow 1$ ;
- 13     **Se**  $\ell(u) = 1$  **então**
- 14         **Para todo**  $x \in N_G(u)$ , **com**  $\ell(x) = 0$  **e**  $a(x) = 0$  **faça**
- 15              $\ell(x) \leftarrow 2$ ;
- 16             enfileira( $x, Q$ );
- 17              $a(x) \leftarrow 1$ ;
- 18     **Senão se**  $\ell(u) = 2$  **então**
- 19         **Para todo**  $y \in N_G(u)$ , **com**  $\ell(y) = 0$  **e**  $a(y) = 0$  **faça**
- 20             enfileira( $y, Q$ );
- 21              $a(y) \leftarrow 1$
- 22 **Retorna**  $\ell$ ;

---

No lema a seguir mostramos que o Algoritmo 1 sempre retornará uma função que rotula  $V(G)$ , observando algumas condições.

**Lema 1.** *O Algoritmo 1 retorna uma função  $\ell : V(G) \rightarrow \{1, 2\}$ , satisfazendo as seguintes condições:*

- (i) *papel 2 possui adjacência com papel 1;*
- (ii) *papel 1 não possui adjacência com ele mesmo;*
- (iii) *para todo  $v \in V(G)$ ,  $\ell(v) \in \{1, 2\}$ .*

*Demonstração.* Queremos mostrar na condição (i) que, ao término da execução do algoritmo, os vértices com papel 2 possuem adjacência com vértices de papel 1.

Consideremos o *loop* da Linha 14 no Algoritmo 1: a condição para se realizar esse *loop* é  $\ell(u) = 1$ , ou seja,  $u$  ter papel 1. Em seguida, dentro do *loop*, para todo vizinho de  $u$  é atribuído papel 2.

Dado que, ao longo deste algoritmo, essa é a única condição para atribuição de papel 2, ou seja, ser adjacente à vértices de papel 1, temos que necessariamente todos os vértices de papel 2 possuem adjacência com papel 1.

Similarmente, a condição (ii) pode ser provada usando o *loop* da Linha 14. Dado que um vértice possui atribuição de papel igual a 1, todos os seus vizinhos receberão atribuição de papel 2 e na sequência enfileirados. Para a atribuição de papel 1 é necessário, segundo a condição da Linha 12, que o vértice desenfileirado em  $u$  não tenha atribuição.

Desta forma, temos que toda a vizinhança de 1 recebe papel 2, garantindo que nenhum vizinho vá para fila sem atribuição e receba posteriormente papel 1. Ou seja, nenhum vértice com atribuição de papel 1 terá adjacência com vértice com papel 1.

Nos resta mostrar que o Algoritmo 1 satisfaz a condição (iii), ou seja, que todos os vértices recebem um rótulo do conjunto  $\{1, 2\}$ .

Temos que as condições dentro do *loop* da Linha 9, garantem que todos os vizinhos desse vértice sejam adicionados à fila, pois sempre que o *loop* é executado é desenfileirado um vértice e a partir desse vértice é adicionada toda sua vizinhança na fila. A condição para execução do *loop* da Linha 9 é enquanto a fila não estiver vazia, ou seja, enquanto todos os vértices presentes nela não forem visitados. A condição das Linhas 11 e 13 garantem que todos os vértices referenciados no *loop* possuam um papel, seja ele 1 ou 2.

Dessa forma temos que todos os vértices são adicionados à fila e que todos recebem um rótulo de  $\{1, 2\}$ . □

A partir da rotulação do grafo  $G$ , é possível definir uma 2- e 3-atribuição para  $G \boxtimes K_2$ . Nas próximas duas subseções apresentamos funções  $p$  e  $q$  de atribuição a partir da função  $\ell$ .

### 3.2. 2-atribuição de papéis em $G \boxtimes K_2$

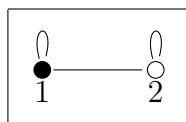


Figura 2. Grafo  $R_1$ .

**Teorema 1.** *Seja  $G$  um grafo não trivial. Então  $G \boxtimes K_2$  possui uma 2-atribuição de papéis.*

*Demonstração.* Definimos  $p : V(G \boxtimes K_2) \rightarrow \{1, 2\}$  como:

Para cada  $v = (v, u_1) \in V(G_2)$ , faça  $p(v) = \ell(v)$  e para todo  $v' = (v, v_2) \in V(G \boxtimes K_2)$ :

$$p(v') = \begin{cases} 1, & \text{se } p(v) = 1; \\ 2, & \text{se } p(v) = 2. \end{cases}$$

Mostramos que  $p$  é uma  $R_1$ -atribuição de papéis.

Como provado no Lema 1 e, dado que  $G$  é não-trivial, em  $G$  todos os vértices de rótulo 1 fazem adjacência com rótulo 2. Assim como todos os vértices de rótulo 2 sempre fazem adjacência com rótulo 1.

Nos resta provar que para todo  $v \in V(G \boxtimes K_2)$ , tal que  $\ell(v) = 1$ , possui vizinhança com papel 1 e para todo  $v \in V(G \boxtimes K_2)$ , tal que  $\ell(v) = 2$ , possui vizinhança com papel 2. Sabemos que para todo  $v \in V(G \boxtimes K_2)$ ,  $vv' \in E(G \boxtimes K_2)$ . Pela definição de  $p$ , temos que se  $p(v) = 1$  então  $p(v') = 1$  e, ainda, se  $p(v) = 2$  então  $p(v') = 2$ .

Dessa forma, mostramos que todo vértice  $v \in V(G \boxtimes K_2)$ , possui  $N_{G \boxtimes K_2}(v) \in \{1, 2\}$ . Portanto,  $p$  é uma  $R_1$ -atribuição de papéis para  $G \boxtimes K_2$ .  $\square$

### 3.3. 3-atribuição de papéis em $G \boxtimes K_2$

**Teorema 2.** *Seja  $G$  um grafo não trivial. Então  $G \boxtimes K_2$  possui uma 3-atribuição de papéis.*

*Demonstração.* Definimos  $q : V(G \boxtimes K_2) \rightarrow \{1, 2, 3\}$  como:

Para cada  $v = (v, u_1) \in V(G_2)$ , faça  $q(v) = \ell(v)$  e para todo  $v' = (v, v_2) \in V(G \boxtimes K_2)$ :

$$q(v') = \begin{cases} 3, & \text{se } q(v) = 1; & (1.1) \\ 2, & \text{se } q(v) = 2. & (1.2) \end{cases}$$

Para concluirmos que  $q$  é uma  $R_2$ -atribuição de papéis, mostramos que, para todo  $v \in V(G \boxtimes K_2)$ :

- se  $q(v) = 1$ , então  $N_{G \boxtimes K_2}(v) = \{2, 3\}$ ;
- se  $q(v) = 2$ , então  $N_{G \boxtimes K_2}(v) = \{1, 2, 3\}$ ;
- se  $q(v) = 3$ , então  $N_{G \boxtimes K_2}(v) = \{1, 2\}$ .

Como provado no Lema 1, dado que  $|V(G)| \geq 2$ , em  $G$  todos os vértices de rótulo 1 fazem adjacência com rótulo 2. Assim como todos os vértices de rótulo 2 sempre fazem adjacência com rótulo 1.

Por (1.2) temos que todos os vértices com papel 2 possuem adjacência com papel 2 e por (1.1) todos os vértices de papel 1 possuem adjacência com papel 3 e vice-versa.

Nos resta mostrar então, que todo vértice de papel 2 possui adjacência com papel 3 e todo vértice de papel 3 possui adjacência com papel 2.

Pela definição de produto forte, temos que se  $vu \in E(G)$ , então em  $G \boxtimes K_2$ ,  $v$  possui adjacência com o correspondente de  $u$ , ou seja  $vu' \in E(G \boxtimes K_2)$ . Da mesma forma, temos que se  $uv \in E(G)$ , então  $u'v' \in E(G \boxtimes K_2)$ .

Dado que  $v \in V(G)$  e  $q(v) = 2$ , temos, pelo Lema 1, que  $v$  faz adjacência com um vértice  $u$  de papel 1 e, pelo produto forte, temos que  $v$  faz adjacência com o correspondente de  $u$ , no caso, pela função  $q$ , o correspondente  $u'$  possui papel 3. Desta forma, temos que todos os vértices que recebem papel 2 possuem adjacência com um vértice que possui papel 3.

Dado que todo vértice correspondente de um vértice  $v$ , de papel 1, recebe papel 3, pelo Lema 1, sabemos que o vértice  $v$  possui adjacência com um vértice  $u$  de papel 2. Logo, temos que  $v'u, v'u' \in E(G \boxtimes K_2)$ , ou seja, um vértice de papel 3 possui adjacência com vértice de papel 2. Temos, desta forma, que todo vértice com papel 3 possui adjacência com vértice de papel 2.

Temos que em  $G$ , um vértice  $v$  com atribuição de papel igual a 1 possui adjacência apenas com vértice 2. Em  $G \boxtimes K_2$ ,  $v$  terá adjacência com seus correspondentes, com atribuição de papel igual a 3 e com os correspondentes de seus vizinhos em  $G$ , cuja atribuição de papel é 2, segundo a definição de  $q$ . Observamos então, que todo  $v \in V(G \boxtimes K_2)$ , cujo  $q(v) = 1$ , não possui adjacência com um vértice de papel 1.

Similarmente, um vértice com atribuição de papel igual a 3 terá adjacência apenas com seu correspondente de papel igual a 1, com os vizinhos do seu correspondente, que pelo Lema 1, possuem atribuição 2 e com os correspondentes dos vizinhos dos seus correspondentes, que por definição de  $q$  recebem papel 2.  $\square$

### 3.4. 2- e 3-atribuição de papéis em $G \boxtimes K_n$ .

Aproveitando as funções  $p$  e  $q$  definidas na demonstração dos Teoremas 1 e 2, obtemos o seguinte resultado.

**Teorema 3.** *Sejam  $G$  um grafo não trivial e  $n \geq 3$ . Então  $G \boxtimes K_n$  possui 2- e 3-atribuição de papéis.*

*Demonstração.* O Teorema 1 mostra que para qualquer grafo não trivial  $G$ , temos uma função  $p$  que nos dá uma 2-atribuição de papel ao produto forte de  $G$  com  $K_2$ . Ao incorporar  $n - 2$  vértices universais ao grafo  $K_2$ , expandimos o grafo resultante,  $G \boxtimes K_n$ , criando  $n - 2$  colunas adicionais, e pelo produto forte, cada vértice em uma coluna está adjacente a todos os vértices na linha correspondente ao vértice no grafo  $K_n$ .

Para obtermos uma  $R_1$ -atribuição para  $G \boxtimes K_n$ , ajustamos a função  $p$  para atribuir os mesmos papéis de  $v \in V(G)$  a todos os vértices nas colunas correspondentes. Cada vértice terá adjacência com vértices de papéis 1 e 2, dado o resultado do Algoritmo 1. Isso implica em uma  $R_1$ -atribuição para  $G \boxtimes K_n$ .

Para definirmos uma  $R_2$ -atribuição, consideramos a função  $q$  para  $G \boxtimes K_2$ . Para  $G \boxtimes K_n$ , adicionamos a seguinte regra à  $q$ : atribuímos o papel 2 para todos os vértices das  $n - 2$  colunas restantes.

Pelo Teorema 2, é provado que os vértices de papel 1 não possuem adjacência com papéis de mesma atribuição, da mesma forma, os vértices de papel 3.

Nos resta mostrar que todos os vértices que possuem atribuição 2 possuem adjacência com papéis 1, 2 e 3.

Por definição de grafo completo, temos que todos os vértices possuem adjacência entre si, desta forma se em  $G \boxtimes K_2$  temos que, pela função  $q$ , em uma linha possuem vértices de papéis 1 e 3, logo, todos os vértices de papel 2 da mesma linha em  $G \boxtimes K_n$  farão adjacência com papéis 1 e 3. Dado que as colunas induzem uma cópia do grafo  $G$  para cada vértice de  $K_n$ , e  $G$  é não trivial, então os vértices de papel 2 farão adjacência com outro vértice, da mesma coluna, que recebe também papel 2.



Para os vértices de atribuição 2 que estão na mesma linha de sequência de atribuição 2 no grafo  $G \boxtimes K_2$ , temos que, por definição de produto forte, farão adjacência com todos os vértices da linha adjacente, que por definição de  $q$  possui vértices com atribuição de papel 1 e 3. Assim, concluímos que  $G \boxtimes K_n$  possui uma  $R_2$ -atribuição de papéis.  $\square$

### 3.5. Complexidade

No Teorema 3 mostramos que  $G \boxtimes K_n$  sempre tem 2- e 3-atribuição de papéis, quando  $G$  é um grafo não trivial. Isso torna a complexidade de se decidir  $r$ -atribuição constante, para  $r = 2, 3$ . Apesar desse resultado, mostramos aqui que tal atribuição pode ser determinada em tempo linear, analisando a complexidade assintótica superior de tempo de execução do Algoritmo 1 e das funções  $p$  e  $q$ .

No Algoritmo 1, consideraremos a frequência com que cada vértice é visitado e desconsideramos as operações constantes para determinar um limite assintótico superior. Definimos  $n = |V(G)|$  e  $m = |E(G)|$ .

Na fase de inicialização, cada vértice é visitado apenas uma vez, o que nos leva a uma complexidade de  $\mathcal{O}(n)$ .

As condições das Linhas 17 e 21 garantem que cada vértice será enfileirado apenas uma vez, assegurando que o *loop* “enquanto” da Linha 9 será executado uma vez para cada vértice, resultando em uma complexidade de  $\mathcal{O}(n)$ .

Da mesma forma, as Linhas 14 e 19 garantem que a adjacência de cada vértice é examinada apenas uma vez. Em uma implementação por Listas de adjacências, a soma dos comprimentos de todas as listas de cada vértice é, portanto,  $\mathcal{O}(m)$ . Logo, temos que a complexidade do Algoritmo 1 é  $\mathcal{O}(n) + \mathcal{O}(n) + \mathcal{O}(m) = \mathcal{O}(n + m)$ .

Para as funções  $p$  e  $q$  temos que a complexidade para executar cada uma é de  $\mathcal{O}(n)$ , pois a atribuição se daria apenas a partir do acesso a cada vértice pertencente a  $V(G)$ .

Desta forma, a complexidade total para executar o Algoritmo 1 e uma das funções,  $p$  ou  $q$ , é de:  $\mathcal{O}(n + m) + \mathcal{O}(n) = \mathcal{O}(n + m)$ .

## 4. Conclusão

Neste artigo, mostramos que o produto forte entre um grafo  $G$ , não trivial, e um grafo completo  $K_n$  sempre possui 2- e 3-atribuição de papéis, o que torna a complexidade do problema constante. Tal resultado é obtido por meio de algoritmos lineares que encontram 2- e 3-atribuição para  $G \boxtimes K_n$ .

Por fim, para trabalhos futuros, sugerimos o estudo da complexidade dos problemas de 2- e 3-atribuição para produto forte envolvendo outras classes de grafos.

## Referências

- [1] J. A. Bondy and U. S. R. Murty. *Graph theory with applications*, volume 290. Citeseer, New York, USA, 1976.
- [2] D. Castonguay, E. S. Dias, and F. N. Mesquita. Prismas complementares com 2-atribuição de papéis. *Matemática Contemporânea*, 46:83–93, 2018.

- [3] D. Castonguay, E. S. Dias, F. N. Mesquita, and J. R. Nascimento. Computing some role assignments of cartesian product of graphs. *RAIRO-Oper. Res.*, 56(1):115–121, 2022.
- [4] D. Castonguay, E. S. Dias, F. N. Mesquita, and J. R. Nascimento. Computing role assignments of cartesian product of graphs. *RAIRO-Oper. Res.*, 57(3):1075–1086, 2023.
- [5] R. Diestel. *Graph theory*. New York, USA, Springer-Verlag, 2000.
- [6] M. C. Dourado. Computing role assignments of split graphs. *Theoretical Computer Science*, 635:74–84, 2016.
- [7] M. G. Everett and S. Borgatti. Role colouring a graph. *Mathematical Social Sciences*, 21(2):183–188, 1991.
- [8] J. Fiala and D. Paulusma. A complete complexity classification of the role assignment problem. *Theoretical computer science*, 349(1):67–81, 2005.
- [9] R. H. Hammack, W. Imrich, and S. Klavžar. *Handbook of product graphs*, volume 2. CRC press Boca Raton, 2011.
- [10] T. R. Jensen and B. Toft. *Graph coloring problems*. John Wiley & Sons, 2011.
- [11] S. Lyle. *Homomorphisms of graphs*. PhD thesis, Clemson University, Clemson, South Carolina, 2008.
- [12] F. N. Mesquita. *Atribuição de papéis em alguns produtos de grafos*. PhD thesis, Universidade Federal de Goiás, 2022.
- [13] S. Pandey. *Role colouring hereditary classes of graphs*. PhD thesis, Indian Institute of Science Education and Research Pune, 2019.
- [14] C. Purcell and P. Rombach. On the complexity of role colouring planar graphs, trees and cographs. *Journal of Discrete Algorithms*, 35:1–8, 2015.
- [15] C. Purcell and P. Rombach. Role colouring graphs in hereditary classes. *Theoretical Computer Science*, 876:12–24, 2021.
- [16] F. S. Roberts and L. Sheng. How hard is it to determine if a graph has a 2-role assignment? *Networks*, 37(2):67–73, 2001.
- [17] P. van 't Hof, D. Paulusma, and J. M. van Rooij. Computing role assignments of chordal graphs. *Theoretical Computer Science*, 411(40):3601–3613, 2010.
- [18] Y.-q. Zhao, W.-l. Feng, H. Li, and J.-m. Yang.  $k$ -role assignments under some graph operations. *Journal of Hebei University of Science and Technology*, page 06, 2010.