




De 3-Coloração a Coloração de Arestas: Contribuições Pedagógicas para o Aprendizado no Escopo da Teoria da Computação

From 3-Coloring to Edge Coloring: Pedagogical Contributions for Learning in the Scope of Computation Theory

Ana Júlia Campos Vieira ¹, Dallyla de Moraes Sousa ¹, Daniel Martins da Silva ² e Tanilson Dias dos Santos ¹

¹ Universidade Federal do Tocantins, Ciência da Computação, Tocantins, Brasil

² Universidade Federal do Norte do Tocantins, Logística, Tocantins, Brasil

Data de recebimento do manuscrito: 03/12/2025

Data de aceitação do manuscrito: 05/01/2026

Data de publicação: 10/02/2026

Resumo—Este artigo analisa a NP-completude do problema Edge Coloring através de uma cadeia de redução polinomial iniciada no problema 3-Coloring. A metodologia utiliza a construção de um Grafo Linha para demonstrar a pertinência à classe NP e estabelece a NP-Dificuldade via redução do 3-SAT, baseada em Holyer. São apresentadas definições formais, contexto histórico e revisão de trabalhos fundamentais em teoria da complexidade computacional. O principal resultado demonstra que Edge Coloring é NP-completo por meio de um método de redução claro e acessível. O trabalho oferece exemplos educacionais com ilustrações visuais e explicações passo a passo sobre gadgets lógicos. Este material serve como recurso de aprendizagem para auxiliar estudantes na compreensão de reduções polinomiais e conceitos de NP-completude em Ciência da Computação.

Palavras-chave—NP-completude, Coloração de Arestas, Coloração de Vértices, Reduções Polinomiais, Complexidade Computacional

Abstract—This article examines the NP-completeness of the Edge Coloring problem through a polynomial reduction chain starting from the 3-Coloring problem. The methodology employs a Line Graph to demonstrate membership in NP and establishes NP-Hardness via reductions from 3-SAT, following Holyer's construction. We present formal definitions, historical context, and a review of fundamental works in computational complexity theory. The main result demonstrates that Edge Coloring is NP-complete using a clear and accessible reduction method. The work provides educational examples with visual illustrations and step-by-step explanations of logical gadgets. This material serves as a learning resource to help students understand polynomial reductions and NP-completeness concepts in computer science courses.

Keywords—NP-completeness, Edge Coloring, Vertex Coloring, Polynomial Reductions, Computational Complexity

I. INTRODUÇÃO

A Teoria da Computação investiga os limites fundamentais dos algoritmos, sendo a classe dos problemas NP-Completo o cerne dos desafios práticos e teóricos da área. O estudo desta classe é essencial para entender a intratabilidade computacional, orientando o desenvolvimento de heurísticas e algoritmos de aproximação para problemas cruciais em otimização e inteligência artificial [1].

Nesse contexto, os problemas de coloração de grafos,

como o 3-Coloring (Coloração de Vértices com 3 cores) e o Edge Coloring (Coloração de Arestas), são centrais. O 3-Coloring consiste em determinar se os vértices de um grafo podem ser coloridos com três cores de forma que vértices adjacentes não compartilhem a mesma cor. Historicamente, ele é um dos primeiros problemas a ter sua NP-completude provada por redução do SAT (Satisfiability) [2]. Já o Edge Coloring questiona se as arestas de um grafo podem ser coloridas com k cores de modo que arestas adjacentes (que compartilham um vértice) tenham cores distintas.

Ambos os problemas, apesar de conceitualmente distintos, compartilham uma estrutura computacional equivalente. Enquanto o 3-Coloring possui aplicações clássicas em planejamento e alocação de frequência, o Edge Coloring é

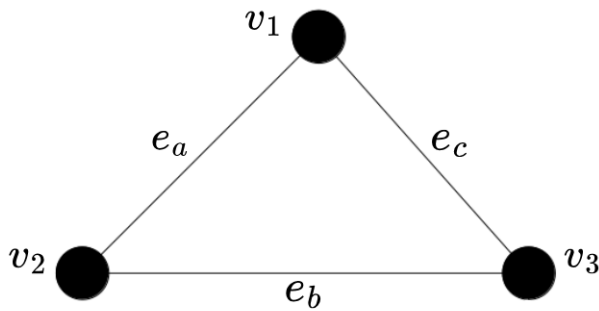


Figura 1: Grafo completo K_3 — exemplo de grafo não-direcionado com 3 vértices.

fundamental em problemas de escalonamento, alocação de recursos em redes e otimização de tempo [3].

Este artigo reúne e organiza demonstrações presentes na literatura sobre a NP-completude do *Edge Coloring*, com foco na clareza conceitual. A redução polinomial de $3\text{-Coloring} \leq_p \text{Edge Coloring}$ é apresentada por meio do conceito de *Grafo Linha*, destacando os elementos centrais da transformação. O objetivo é oferecer um material que apoie o estudo das técnicas de redução e sua importância dentro da Teoria da Computação.

O trabalho expõe a prova e discute o raciocínio envolvido na construção, enfatizando aspectos que contribuem para o ensino de complexidade computacional. As seções seguintes apresentam os fundamentos necessários, a descrição da redução e as reflexões que surgem a partir dessa análise.

II. PRELIMINARES

Começamos revisitando alguns conceitos fundamentais sobre grafos e problemas de coloração, essenciais para compreender o desenvolvimento deste trabalho.

Um *Grafo Não-Direcionado* é definido como uma estrutura $G = (V, E)$, onde V representa um conjunto finito e não vazio de vértices, e $E \subseteq \{(u, v) \mid u, v \in V, u \neq v\}$ é o conjunto de arestas que conectam esses vértices. Neste tipo de grafo, as arestas não possuem orientação, o que significa que a aresta (u, v) é idêntica à aresta (v, u) , estabelecendo uma relação simétrica entre os vértices. Grafos não-direcionados são particularmente úteis para modelar relações mútuas, como amizades em redes sociais, conexões entre computadores em uma rede ou relações de adjacência em mapas.

Dentro dessa estrutura, dizemos que dois *vértices são adjacentes* se existe uma aresta conectando-os diretamente. Esta relação de adjacência é fundamental para definir a estrutura do grafo e suas propriedades. No grafo K_3 (Figura 1), todos os vértices são adjacentes entre si, formando um triângulo completo onde cada vértice possui grau 2 (duas conexões). O conjunto de vértices adjacentes a um vértice v é denominado sua *vizinhança*. A adjacência é uma relação binária que determina a conectividade direta no grafo, sendo essencial para definir caminhos, ciclos e outras propriedades estruturais.

Da mesma forma, duas *arestas são incidentes* quando compartilham um vértice em comum. Esta relação de incidência conecta o conceito de vértices com o conceito

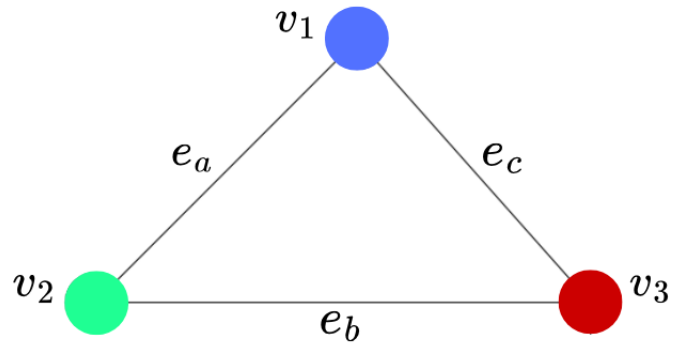


Figura 2: Grafo exemplo para 3-Coloring — estrutura com restrições de adjacência que permite coloração com 3 cores.

de arestas, criando a estrutura combinatória do grafo. No exemplo da Figura 1, qualquer par de arestas entre e_a , e_b e e_c compartilha um vértice, o que as torna incidentes entre si. Uma aresta é dita *incidente* a um vértice quando este vértice é uma de suas extremidades. O *grau* de um vértice é definido como o número de arestas incidentes a ele, sendo esta uma medida fundamental da centralidade do vértice no grafo.

Uma *coloração de vértices* é uma atribuição de cores aos vértices de um grafo por meio de uma função $c : V \rightarrow C$, onde C representa um conjunto finito de cores disponíveis. A notação $c(v)$ indica a cor atribuída ao vértice v , isto é, o resultado da função quando aplicada a esse vértice. Dessa forma, cada vértice recebe exatamente uma cor, permitindo analisar propriedades estruturais do grafo a partir dessa atribuição.

Uma coloração é dita *própria* quando nenhum par de vértices adjacentes compartilha a mesma cor. Em termos formais, isso significa que, para toda aresta $(u, v) \in E$, deve valer $c(u) \neq c(v)$. A condição $c(u)$ e $c(v)$ serem diferentes garante que vértices conectados não entrem em conflito de cor, constituindo o requisito fundamental em problemas clássicos como o 3-Coloring.

A Figura 2 apresenta um grafo simples usado para ilustrar relações de adjacência e incidência em um contexto de coloração. O exemplo evidencia como diferentes conexões afetam as possibilidades de coloração de vértices e de arestas.

O conceito de *Grafo Linha* estabelece uma dualidade entre vértices e arestas. Dado um grafo $G = (V, E)$, seu grafo linha $L(G) = (V_L, E_L)$ é construído mapeando cada aresta em E para um vértice em V_L , e dois vértices em $L(G)$ são adjacentes se as arestas correspondentes em G compartilham um vértice [4]. Esta transformação permite analisar propriedades das arestas do grafo original através do estudo dos vértices do grafo linha.

No exemplo da Figura 3, o grafo linha de K_3 é isomorfo ao próprio K_3 , ilustrando como a transformação preserva a estrutura de adjacência. Isso acontece porque, em K_3 , todo par de arestas compartilha um vértice. Por exemplo, e_a e e_b são adjacentes no grafo linha porque compartilhavam v_2 no grafo original; o mesmo ocorre para os outros pares. Essa correspondência direta entre incidência e adjacência é o que torna o grafo linha uma ferramenta tão útil em reduções entre problemas de coloração.

No contexto de problemas de coloração, dois se destacam:

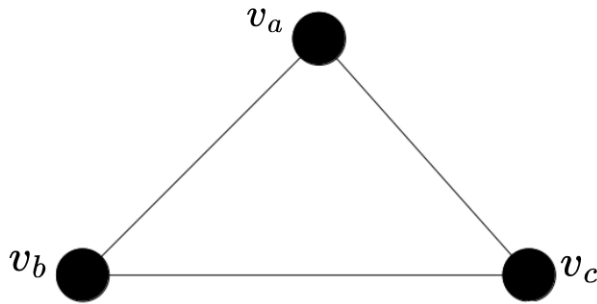


Figura 3: Grafo linha $L(K_3)$ — arestas do grafo original tornam-se vértices, e a incidência transforma-se em adjacência.

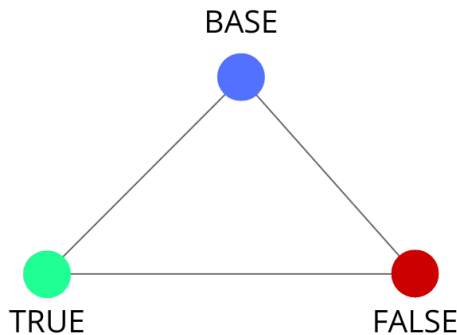


Figura 4: Gadget baseado em K_3 , usado para impor restrições de coloração em reduções para 3-Coloring.

o 3-Coloring (Coloração de Vértices com 3 cores) e o Edge Coloring (Coloração de Arestas). O primeiro é um problema de decisão que busca determinar se existe uma função $c : V \rightarrow \{1, 2, 3\}$ tal que, para toda aresta $(u, v) \in E$, tenhamos $c(u) \neq c(v)$ [5, 6]. Uma coloração que satisfaz esta propriedade é chamada de *coloração própria*. A restrição de usar apenas três cores torna este problema particularmente desafiador, já que muitos grafos exigem mais cores para uma coloração própria, enquanto outros podem ser coloridos com menos.

Os *gadgets* desempenham um papel central nas reduções polinomiais envolvendo problemas de coloração. Um gadget é um pequeno subgrafo construído para impor restrições locais sobre as escolhas de cor, funcionando como um "componente lógico" dentro de reduções maiores. Esse conceito aparece na técnica clássica de *Component Design*, frequentemente utilizada em reduções para o problema 3-Coloring.

O triângulo K_3 é um dos gadgets mais utilizados, pois sua estrutura força necessariamente três cores distintas, representando escolhas mutuamente exclusivas — como valores lógicos *TRUE*, *FALSE* e um estado base. A Figura 4 ilustra essa construção didática amplamente adotada em reduções clássicas.

Já o Edge Coloring (Figura 5) pergunta se é possível colorir as arestas de G usando até k cores, por meio de uma função $c : E \rightarrow \{1, 2, \dots, k\}$, de forma que arestas incidentes recebam cores diferentes. O número mínimo de cores necessárias para essa coloração é o *índice cromático*,

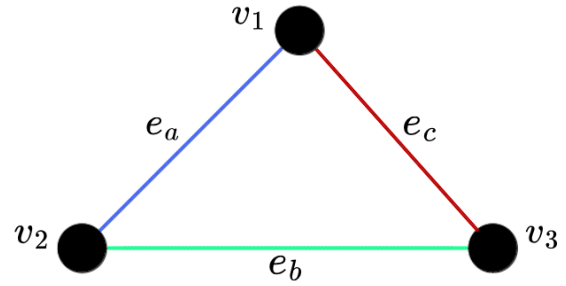


Figura 5: Edge Coloring válido no K_3 — cada aresta recebe uma cor distinta: e_a (azul), e_b (verde), e_c (vermelho).

denotado por $\chi'(G)$. Este problema tem aplicações práticas em escalonamento de tarefas, alocação de frequências em redes wireless e design de torneios esportivos.

No âmbito da complexidade computacional, destacamos duas definições centrais. Uma linguagem de decisão L pertence à *classe NP-Completo* se L está em \mathcal{NP} e toda linguagem em \mathcal{NP} pode ser reduzida a L em tempo polinomial [1, 2]. As reduções são formalizadas pelo conceito de *Redução Polinomial*, onde um problema A é redutível a um problema B (denotado por $A \leq_p B$) se existe uma transformação computável em tempo polinomial que preserva respostas entre instâncias dos dois problemas. Esta noção de redução é a base para estabelecer relações de dificuldade entre problemas e para construir hierarquias de complexidade.

A importância das reduções polinomiais vai além do aspecto teórico, pois elas fornecem informações sobre a estrutura dos problemas e permitem que algoritmos desenvolvidos para um problema sejam adaptados para outros. No contexto educacional, compreender essas reduções é essencial para desenvolver uma intuição sobre quais problemas são computacionalmente difíceis e por quê.

III. TRABALHOS RELACIONADOS

No trabalho de Cook [2], estabeleceu-se a base da NP-completude. Na pesquisa, mostrou-se como verificar soluções em tempo polinomial e construiu a primeira redução polinomial para o SAT. O objetivo foi entender quando um problema permite conferir respostas em pouco tempo. Os resultados são interessantes por terem aberto caminho para Karp ampliar essa ideia, relacionando vários problemas clássicos, incluindo o 3-Coloring, e mostrando como muitos deles caem na mesma classe de complexidade.

No trabalho clássico de Garey e Johnson [1], é detalhada a complexidade do *Graph 3-Colorability*. Na obra, os autores organizam a teoria da NP-completude e utilizam a técnica de *Component Design*, que consiste em montar grafos a partir de peças pequenas que impõem restrições locais. Gadgets como o triângulo K_3 mostram como estruturas pequenas conseguem impor escolhas de cor e controlar o comportamento local do grafo, simulando literais e cláusulas. Os resultados são interessantes por padronizar as reduções que conectam problemas centrais da computação.

No survey escrito por Cao e outros autores [7], podemos notar uma visão geral desse tema. Na pesquisa, os autores reúnem resultados sobre algoritmos, limites estruturais,

casos especiais e questões abertas da coloração de arestas. O objetivo do survey é organizar o que já se sabe sobre o problema, desde técnicas simples de recoloração até métodos mais atuais. Os resultados são interessantes por consolidar o conhecimento disperso sobre limites superiores e conjecturas da área.

No trabalho de Holyer [3], surge o resultado que estrutura a base moderna dessa área. Na pesquisa, o autor apresenta a primeira prova de NP-completude do *Edge Coloring*, mostrando que o problema permanece intratável mesmo quando restrito a grafos cúbicos. A técnica usada envolve montar blocos que forcem escolhas de cor que se propagam pelo grafo inteiro. Os resultados são interessantes por revelar que o índice cromático capta decisões combinatórias fortes e que o problema não se resume a uma variação simples da coloração de vértices.

No trabalho de Basavaraju e Chandran [8], há um resultado importante para a coloração de arestas acíclica em grafos planares. Os autores demonstram que todo grafo planar admite tal coloração com $\Delta(G) + 12$ cores, superando o limite anterior de $2\Delta(G) + 29$. A prova utiliza configurações inevitáveis em grafos planares e trocas de cores para evitar ciclos bicromáticos, mostrando como o índice cromático acíclico reflete a estrutura desses grafos.

No estudo de Galby, Lima, Paulusma e Ries [9], trabalhos mais novos reforçam essa visão. Na pesquisa, o objetivo foi classificar o k -Edge Coloring para grafos H -livres, combinando reduções com análise estrutural para mapear quando o problema é polinomial.

No trabalho de Sinnamoni [10], são propostos algoritmos para coloração de arestas que buscam ser acessíveis e eficientes. O objetivo da autora é desenvolver métodos simples e rápidos para produzir colorações com $d + 1$ cores, onde d é o grau máximo do grafo. A técnica utilizada se baseia em decomposição recursiva e ciclos de Euler para agilizar o processo. Os resultados demonstram que é possível obter boas soluções de forma prática para grafos gerais em aplicações reais.

Por fim, no trabalho de Raeisi e Gholami [11], a coloração de arestas é aplicada à construção de grafos Tanner livres de ciclos curtos para códigos LDPC de peso-coluna três, melhorando a decodificação em canais ruidosos. O método utiliza algoritmos de coloração eficientes para garantir propriedades acíclicas nos grafos bipartidos. Os resultados conectam a combinatória gráfica a aplicações práticas em comunicações digitais.

IV. DESCRIÇÃO DO PROBLEMA

Nesta seção, definimos formalmente os três problemas que aparecem na cadeia de redução. A Tabela 1 apresenta a definição do problema 3-SAT, a Tabela 2 descreve o problema 3-Coloring e a Tabela 3 formaliza o problema Edge Coloring, indicando, em cada caso, a entrada e a pergunta associadas.

TABELA 1: DEFINIÇÃO DO PROBLEMA 3-SAT

3-SAT (Satisfatibilidade Booleana)

Entrada: um conjunto X de variáveis; uma coleção C de cláusulas sobre X onde, para cada $c \in C$, a cláusula possui exatamente 3 literais ($|c| = 3$).

Pergunta: Determinar se existe uma atribuição de valores às variáveis em X de modo que cada cláusula em C tenha pelo menos um literal verdadeiro.

Para fins de formalização, define-se um *literal* como uma variável booleana (x) ou sua negação ($\neg x$). Uma *cláusula* é composta pela disjunção lógica (operador OU) desses literais. A especificidade do problema 3-SAT reside na estrutura rígida onde cada cláusula deve conter estritamente três literais, o que permite a padronização dos componentes gráficos (gadgets) utilizados na redução.

A Tabela 1 define o 3-SAT, que serve como o elo de conexão fundamental nesta prova. Diferente dos problemas de coloração, que lidam com estruturas gráficas, o 3-SAT lida com lógica pura. A restrição de ter "exatamente três literais" é o que permite criar padrões geométricos fixos (como triângulos) nas reduções para grafos.

TABELA 2: DEFINIÇÃO DO PROBLEMA 3-COLORING

3-COLORAÇÃO (3-COLORING)

Entrada: Um grafo $G = (V, E)$.

Pergunta: Existe uma função $c : V \rightarrow \{1, 2, 3\}$ tal que vértices adjacentes recebam cores distintas?

A Tabela 2 define formalmente o problema 3-Coloring. A entrada é um grafo qualquer, e a pergunta questiona se é possível colorir seus vértices usando apenas três cores, respeitando a regra básica de que vértices conectados por uma aresta devem ter cores diferentes.

TABELA 3: DEFINIÇÃO DO PROBLEMA EDGE COLORING

COLORAÇÃO DE ARESTAS (EDGE COLORING)

Entrada: Um grafo $G = (V, E)$ e um inteiro k .

Pergunta: Existe uma função $c : E \rightarrow \{1, 2, \dots, k\}$ tal que arestas incidentes recebam cores distintas?

A Tabela 3 define o problema Edge Coloring. Aqui, a entrada inclui um grafo e também um número k que representa a quantidade de cores disponíveis. A pergunta busca saber se podemos colorir as arestas do grafo de forma que arestas que compartilham um vértice comum recebam cores diferentes. Note que enquanto no 3-Coloring colorimos

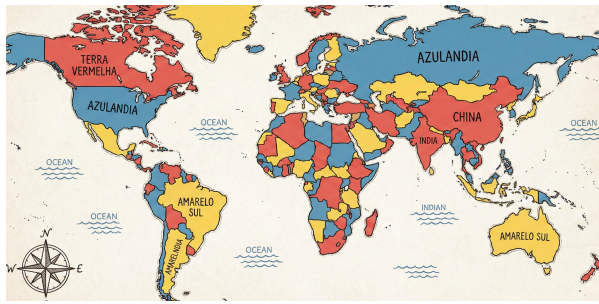


Figura 6: Mapa ilustrando a restrição de adjacência no 3-Coloring.



Figura 7: Ilustração de uma festa como analogia para o 3-SAT.

vértices, no Edge Coloring colorimos arestas, mas ambos impõem restrições de adjacência.

Os problemas de coloração de grafos representam uma classe fundamental na teoria da computação, com aplicações que vão desde o planejamento de horários até a alocação de recursos em sistemas distribuídos. Nesta seção, descrevemos formalmente os dois problemas centrais deste trabalho: o 3-Coloring, um problema clássico de coloração de vértices, e o Edge Coloring, seu análogo na coloração de arestas. Ambos são problemas de decisão NP-completos [1, 2, 3], mas cada um apresenta desafios próprios.

Para tornar esses conceitos mais próximos do cotidiano, podemos imaginar o 3-Coloring como o ato de colorir um mapa usando apenas três cores, garantindo que países vizinhos nunca compartilhem a mesma cor, como ilustrado na Figura 6.

Em relação ao 3-SAT, é como organizar uma festa onde o sucesso depende de satisfazer a todos. Para isso, o organizador deve fazer várias escolhas binárias (as variáveis), como definir se "Haverá Música Ao Vivo?" (Sim ou Não). Cada convidado importante impõe uma cláusula: ele só vem à festa se pelo menos uma de suas três condições for atendida. Por exemplo, um convidado pode exigir: "Eu vou se tiver Música Ao Vivo OU se o Amigo X não vier OU se o Buffet for vegetariano." O desafio 3-SAT é encontrar uma única combinação de decisões (uma configuração Sim/Não para todos os fatores) que satisfaça a exigência de todos os convidados simultaneamente. Se essa combinação existir, a festa pode ser realizada. Essa analogia é ilustrada na Figura 7.

Já o Edge Coloring lembra a montagem da grade de horários de uma escola: arestas representam aulas e vértices representam professores ou salas. Aulas que usam o mesmo recurso não podem ocorrer no mesmo horário, e por isso precisam de cores diferentes, conforme ilustrado na Figura 8.

Essas analogias destacam como problemas abstratos da computação surgem em situações reais.

	Horário	Turma	Aula	Sala	Professor
8h-10h		Turma A	Ciências	Sala 101	Prof. Santos
10h-12h		Turma B	Ciências	Sala 101	Prof. Santos

Figura 8: Grade escolar ilustrando o Edge Coloring.

V. DEMONSTRAÇÃO E CONTRIBUIÇÕES

Para provar que Edge Coloring é NP-Completo é necessário demonstrar duas condições:

- Edge Coloring \in NP
- Edge Coloring \in NP-Difícil

I) Edge Coloring \in NP: Para demonstrar que o problema pertence a \mathcal{NP} , apresentamos um certificado de tamanho polinomial e um algoritmo verificador determinístico capaz de validar esse certificado em tempo polinomial [12].

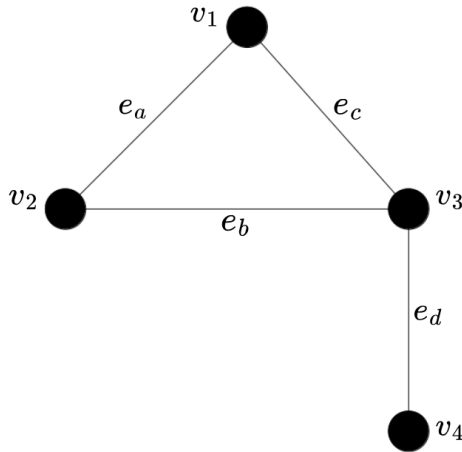
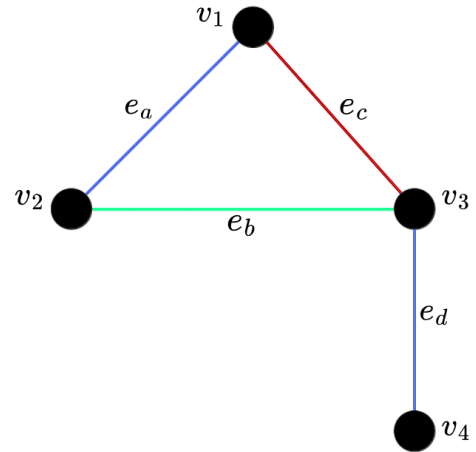
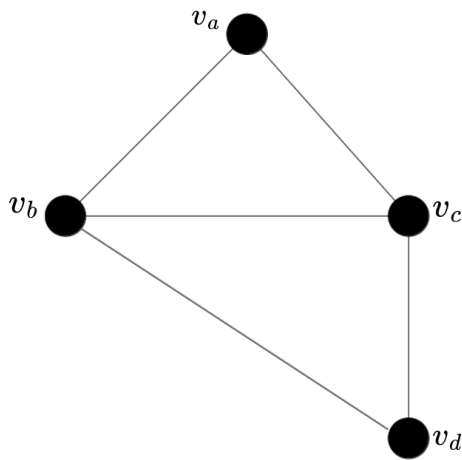
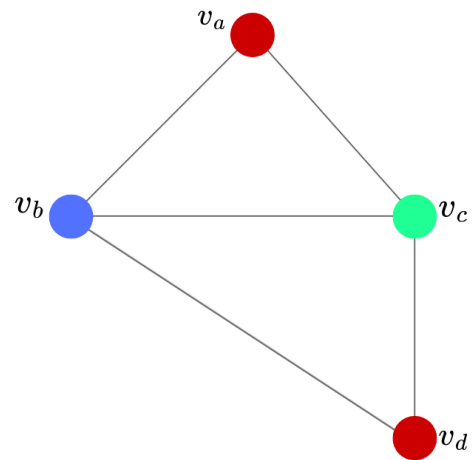
Dado o seguinte certificado: uma coloração $c: E \rightarrow 1, 2, \dots, k$. Um verificador examina todos os pares distintos de arestas do grafo. Para cada par (e_1, e_2) , o algoritmo testa se elas compartilham um vértice (são incidentes) e, caso positivo, confirma que $c(e_1) \neq c(e_2)$. Como existem no máximo $|E|^2$ pares de arestas para verificar, o custo computacional é limitado por $O(|E|^2)$, o que garante a execução em tempo polinomial.

II) Equivalência Estrutural (Grafo Linha): Uma forma pedagógica de visualizar a pertinência a \mathcal{NP} é através da transformação para o Grafo Linha $L(G)$. Essa construção demonstra que o problema de Coloração de Arestas pode ser modelado como um problema de Coloração de Vértices (que sabemos estar em \mathcal{NP}). A transformação mapeia cada aresta de G em um vértice de $L(G)$, e as adjacências entre arestas incidentes em G tornam-se arestas entre vértices em $L(G)$. Assim, uma coloração própria das arestas de G corresponde diretamente a uma coloração própria dos vértices de $L(G)$. Uma vez que o Grafo Linha pode ser construído em tempo polinomial e a coloração de vértices é um problema bem conhecido em \mathcal{NP} , esta equivalência estrutural reforça a classificação do problema de Edge Coloring como pertencente à classe \mathcal{NP} .

Dado um grafo $G = (V, E)$, construímos $L(G) = (V_L, E_L)$ onde:

- $V_L = E$ (cada aresta de G torna-se um vértice em $L(G)$)
 - $E_L = \{(e_1, e_2) \mid e_1, e_2 \in E \text{ são incidentes em } G\}$
- Como exemplo ilustrativo, a Figura 9 mostra um grafo G_{ex} com vértices v_1, v_2, v_3, v_4 e arestas e_a (ligando v_1-v_2), e_b (v_2-v_3), e_c (v_1-v_3) e e_d (v_3-v_4). Observa-se que as arestas e_b e e_d compartilham o vértice v_3 , o que configura uma relação de incidência direta. Por sua vez, e_b também é incidente com e_c , uma vez que ambas incidem em v_3 . Essas relações de adjacência entre arestas no grafo original serão representadas como arestas no grafo linha $L(G_{ex})$.

A Figura 10 mostra o grafo linha correspondente $L(G_{ex})$, onde cada aresta do grafo original torna-se um vértice. A adjacência entre v_b e v_d reflete diretamente o compartilhamento do vértice v_3 pelas arestas e_b e e_d no grafo original.

Figura 9: Grafo Base (G_{ex})Figura 11: Coloração de arestas em G_{ex} Figura 10: Grafo Linha ($L(G_{ex})$)Figura 12: Coloração Válida em $L(G_{ex})$

A propriedade fundamental, conforme Jensen e Toft [13], é que G admite uma k -coloração de arestas se e somente se $L(G)$ admite uma k -coloração de vértices. Esta equivalência confirma que resolver Edge Coloring é redutível a resolver Vertex Coloring, reforçando sua pertinência à classe \mathcal{NP} .

a. Prova de Corretude da Equivalência

A seguir, apresenta-se a demonstração de que G é 3-aresta-colorível se e somente se $L(G)$ é 3-vértice-colorível.

Direção 1(\Rightarrow): Se G é 3-aresta-colorível, então $L(G)$ é 3-vértice-colorível.

Prova: Seja $c : E(G) \rightarrow \{\text{vermelho}, \text{azul}, \text{verde}\}$ uma coloração própria das arestas de G . Para cada aresta $e \in E(G)$, atribuímos a cor $c(e)$ ao vértice correspondente $v_e \in V(L(G))$. Se duas arestas e_b e e_c são incidentes ao mesmo vértice v_3 em G , elas possuem cores diferentes, como ilustrado na Figura 11. Logo, os vértices v_b e v_c em $L(G)$, que são adjacentes, receberão cores diferentes.

Direção 2(\Leftarrow): Se $L(G)$ é 3-vértice-colorível, então G é 3-aresta-colorível.

Prova: Seja $c' : V(L(G)) \rightarrow \{1, 2, 3\}$ uma coloração própria dos vértices de $L(G)$, como mostrado na Figura 12. Definimos a coloração das arestas de G tal que $c(e) = c'(v_e)$. A preservação da adjacência garante que arestas incidentes em G terão cores distintas, validando a solução.

III) *Edge Coloring* \in NP-Difícil: A demonstração da

dificuldade deste problema usa o princípio da transitividade das reduções polinomiais. Para mostrar que Edge Coloring é tão difícil quanto o 3-Coloring, construímos uma cadeia de redução em duas etapas. Primeiro, reduzimos o 3-Coloring para o 3-SAT, transformando restrições de cores em cláusulas booleanas. Depois, reduzimos o 3-SAT para o Edge Coloring, simulando variáveis e cláusulas com um grafo adequado. Pela transitividade, se 3-Coloring é NP-completo, então Edge Coloring é NP-difícil.

A primeira etapa consiste na redução polinomial 3-Coloring \leq_p 3-SAT. Dado um grafo $G = (V, E)$ com n vértices, constrói-se uma fórmula lógica ϕ que codifica de forma precisa as restrições necessárias para uma coloração própria de G com três cores [1]. Para cada vértice $v_i \in V$, são criadas três variáveis booleanas: $x_{i,1}$ (representando a cor 1), $x_{i,2}$ (cor 2) e $x_{i,3}$ (cor 3).

A fórmula ϕ é formada pela conjunção de dois tipos de cláusulas, que juntas modelam as condições de uma coloração própria. A garantia de cor única exige que cada vértice v_i receba pelo menos uma cor, representada pela cláusula $(x_{i,1} \vee x_{i,2} \vee x_{i,3})$. Além disso, a restrição de adjacência assegura que, para cada aresta $(v_i, v_j) \in E$, vértices adjacentes não compartilhem a mesma cor, o que é modelado por três cláusulas de conflito para cada aresta: $(\neg x_{i,1} \vee \neg x_{j,1})$, $(\neg x_{i,2} \vee \neg x_{j,2})$ e $(\neg x_{i,3} \vee \neg x_{j,3})$. Essas cláusulas lógicas correspondem diretamente à restrição de exclusão mútua representada no gadget da Figura 4, demonstrando como condições combinatórias são traduzidas em restrições booleanas.

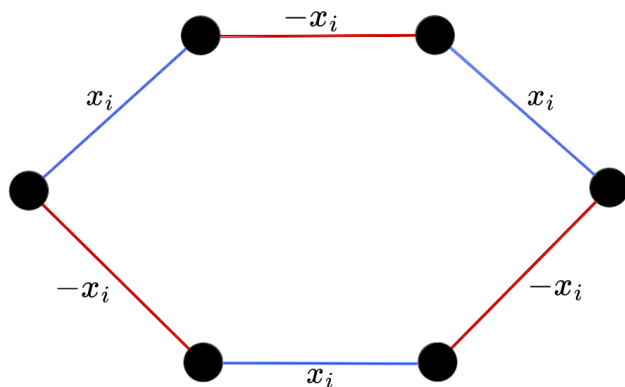


Figura 13: Esquema do Gadget de Variável: a alternância de cores no ciclo simula a negação lógica (x vs $\neg x$).

Uma vez obtida a fórmula satisfatível, avançamos para a redução final para Edge Coloring. Esta etapa baseia-se na construção clássica de Ian Holyer[3], que provou ser NP-completo determinar se o índice cromático de um grafo cúbico é 3 ou 4. A redução converte a fórmula lógica em um grafo cúbico utilizando componentes modulares específicos, denominados *gadgets*, que transportam valores de verdade através de pares de arestas coloridas.

A construção utiliza duas estruturas principais que podem ser compreendidas esquematicamente. A primeira é o *Componente de Variável*, que funciona como um gerador de verdade. Conforme ilustrado conceitualmente na Figura 13, ele é constituído por um ciclo de arestas. Devido à natureza da coloração de arestas, as cores devem se alternar obrigatoriamente ao longo do ciclo. Se associarmos uma cor ao valor "Verdadeiro" e outra ao "Falso", essa alternância garante a consistência lógica: sempre que uma aresta representa x , a adjacente representará $\neg x$.

O segundo elemento crítico é o *Componente de Cláusula*, que atua como um testador de satisfação. Este subgrafo conecta-se às arestas dos ciclos das variáveis correspondentes aos literais da cláusula. Sua propriedade topológica fundamental, representada na Figura 14, é o bloqueio condicional: o gadget é desenhado de tal forma que se torna impossível de colorir com apenas 3 cores se, e somente se, todas as suas arestas de entrada carregarem a cor correspondente ao valor "Falso".

Dessa forma, o grafo cúbico resultante completo só será 3-aresta-colorível se existir uma atribuição de verdade que satisfaça a fórmula 3-SAT original, evitando o conflito nos gadgets de cláusula. Conclui-se, assim, que resolver Edge Coloring é suficiente para resolver 3-SAT e, por transitividade, o 3-Coloring.

Ressalta-se que o resultado em um grafo cúbico fortalece a conclusão. Na teoria da complexidade, se um problema é NP-Difícil para uma classe restrita de entradas (grafos 3-regulares), ele mantém essa propriedade para o caso geral (grafos arbitrários), visto que a classe restrita compõe um subconjunto do problema global, conforme o princípio da restrição descrito por Garey e Johnson [1]. Dessa forma, a prova de Holyer fundamenta a classificação do problema de Edge Coloring como NP-Difícil.

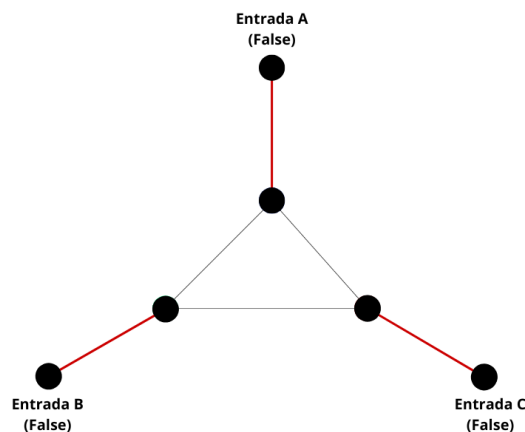


Figura 14: Lógica do Gadget de Cláusula: o componente falha (não é colorível) apenas se receber "Falso" em todas as entradas.

Conclusão Geral: Por (I) e (II) demonstramos a pertinência a \mathcal{NP} . Por (III) justificamos a dificuldade via redução transitiva baseada em Holyer. Logo, Edge Coloring é NP-Completo.

b. Contribuições Pedagógicas

Este trabalho apresenta contribuições ao ensino de Teoria da Computação ao esclarecer a distinção entre equivalência estrutural e redução de dificuldade. A utilização do Grafo Linha permite demonstrar que o problema de Coloração de Arestas pode ser modelado como um problema de Coloração de Vértices, o que comprova sua pertinência à classe \mathcal{NP} conforme as definições de Garey e Johnson [1]. Em contrapartida, a prova de dificuldade exige a construção de componentes lógicos, ou gadgets, como estabelecido por Holyer [3], evitando a confusão comum sobre a direção das reduções polinomiais.

A visualização da transformação estrutural $G \rightarrow L(G)$ [4], complementada por representações visuais, facilita a compreensão geométrica do processo. As ilustrações auxiliam os estudantes a visualizar conceitos abstratos, permitindo o entendimento das relações entre problemas de coloração e a lógica de satisfatibilidade booleana.

VI. RESULTADOS E REFLEXÕES

A análise estrutural via Grafo Linha evidencia como problemas de coloração de naturezas distintas (vértices e arestas) compartilham uma base computacional comum [4]. Esta relação reforça o conceito de que problemas diferentes podem pertencer à mesma classe de complexidade. A equivalência demonstrada confirma que uma instância de coloração de arestas possui solução se, e somente se, a instância correspondente de coloração de vértices no grafo linha também for solucionável.

Uma reflexão central deste estudo recai sobre a complexidade da prova de NP-Dificuldade. Inicialmente, a intuição geométrica sugere uma tentativa de redução direta entre os problemas de coloração. No entanto, a investigação teórica revelou que a redução padrão $3\text{-Coloring} \leq_p \text{Edge Coloring}$ não é imediata em termos de construção de gadgets topológicos diretos. Foi necessário compreender que a

literatura estabelece essa conexão através de uma ponte lógica: a redução transitiva passando pelo problema 3-SAT.

Essa descoberta pedagógica é valiosa: ela demonstra que, embora problemas de grafos sejam visualmente similares, a prova de sua dificuldade muitas vezes exige o retorno aos fundamentos da lógica booleana. A construção de Holyer [3], utilizada neste trabalho, ilustra precisamente como restrições locais em um grafo cúbico simulam portas lógicas, confirmando a intratabilidade do problema mesmo em estruturas restritas.

Sob uma perspectiva prática, a confirmação da NP-completude do Edge Coloring para $k = 3$ indica a necessidade de abordagens alternativas para a solução exata em casos gerais [3]. Essa constatação direciona a investigação para o uso de heurísticas e a análise de casos especiais tratáveis, como os grafos bipartidos, onde o Teorema de Vizing assegura que o índice cromático iguala o grau máximo [14]. Para o ensino, este resultado demonstra que a classificação de complexidade orienta a escolha de estratégias algorítmicas.

A contribuição pedagógica deste trabalho reside na integração de técnicas de redução com suporte visual. A construção do Grafo Linha e a explicação dos componentes de Holyer tornam a demonstração acessível para estudantes de graduação. A apresentação sequencial dos conceitos permite a compreensão dos passos necessários para estabelecer a NP-Completeness de um problema, desde a verificação via equivalência estrutural até a prova de dificuldade via satisfatibilidade lógica.

Os resultados destacam o Grafo Linha como ferramenta pedagógica na teoria da computação [15]. Esta estrutura facilita a compreensão das reduções polinomiais e serve como ponte conceitual entre diferentes áreas da teoria dos grafos. A metodologia adotada pode ser aplicada ao ensino de outros tópicos, combinando formalismo matemático com exemplos visuais para tornar conceitos abstratos tangíveis.

VII. CONSIDERAÇÕES FINAIS

Este artigo estabeleceu a classificação do problema Edge Coloring como NP-Completo mediante uma abordagem dupla. A pertinência a \mathcal{NP} foi demonstrada através da equivalência estrutural com o problema de Coloração de Vértices via Grafo Linha, conforme teoria de Whitney [4]. A condição de NP-Dificuldade foi justificada pela redução polinomial a partir do problema 3-SAT, utilizando a construção de gadgets proposta por Holyer [3], o que valida a relação transitiva com o problema 3-Coloring.

As contribuições pedagógicas compreendem a formalização da prova de redutibilidade e a distinção metodológica entre verificação e prova de dificuldade. A incorporação de exemplos ilustrativos e a discussão sobre as implicações práticas da intratabilidade computacional visam apoiar o aprendizado. O material pode integrar cursos de teoria da computação como exemplo de técnicas de redução e análise de complexidade.

O trabalho demonstra a viabilidade de apresentar conceitos de teoria da computação de maneira compreensível para estudantes de graduação. A abordagem baseada em exemplos visuais constrói a intuição sobre reduções polinomiais e NP-completude [1].

Para o ensino de complexidade computacional, este recurso combina rigor teórico com acessibilidade. A estrutura apresentada permite acompanhar o processo de redução polinomial, desde a transformação inicial até a prova de correteza, o que desenvolve a compreensão dos fundamentos da teoria da NP-completude [2].

Trabalhos futuros podem investigar variantes do problema, como Edge Coloring em classes restritas de grafos ou desenvolver materiais interativos para visualização de reduções. A criação de recursos adicionais como vídeos explicativos ou simulações poderia complementar o material apresentado, ampliando o impacto educacional na área de complexidade computacional.

REFERENCES

- [1] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman and Company, 1979.
- [2] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. ACM, 1971, pp. 151–158.
- [3] I. Holyer, "The np-completeness of edge-coloring," *SIAM Journal on Computing*, vol. 10, no. 4, pp. 718–720, 1981.
- [4] H. Whitney, "Congruent graphs and the connectivity of graphs," *American Journal of Mathematics*, vol. 54, no. 1, pp. 150–168, 1932.
- [5] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*. New York: North Holland, 1976.
- [6] C. H. Papadimitriou, *Computational Complexity*. Reading, Massachusetts: Addison-Wesley, 1994.
- [7] K. Cao, M. Wang, and W. Zhang, "Graph edge coloring: A survey," *arXiv*, 2018, arXiv:1810.08027.
- [8] M. Basavaraju and L. S. Chandran, "Acyclic edge-coloring of planar graphs," *SIAM Journal on Discrete Mathematics*, vol. 23, no. 3, pp. 1618–1627, 2009.
- [9] E. Galby, C. Lima, D. Paulusma, and B. Ries, "Classifying k-edge-colouring for h-free graphs," *arXiv*, 2019, arXiv:1907.03201.
- [10] L. Sinnamon, "Fast and simple edge-coloring algorithms," *arXiv*, 2021, arXiv:2103.01311.
- [11] G. R. Raeisi and M. R. Gholami, "Edge coloring of graphs with applications in coding theory," *China Communications*, vol. 18, no. 1, pp. 181–195, 2021.
- [12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA: MIT Press, 2009.
- [13] T. R. Jensen and B. Toft, *Graph Coloring Problems*. New York: Wiley-Interscience, 1995.

- [14] V. G. Vizing, "On an estimate of the chromatic class of a p-graph," *Diskret. Analiz*, vol. 3, pp. 25–30, 1964, (em Russo).
- [15] P. G. H. Lehot, "An optimal algorithm to detect a line graph and output its root graph," *Journal of the ACM*, vol. 21, no. 4, pp. 569–575, 1974.

