Hitting Set: Contribuições Pedagógicas para o Aprendizado no Escopo da Teoria da Computação

Hitting Set: Pedagogical Contributions for Learning within the Scope of Theory of Computation

João Pedro Melo Póvoa[1], Benedito Jaime Melo Moraes Junior[1], Daniel Martins da Silva[2] e Tanilson Dias dos Santos[1]

[1] *Universidade Federal do Tocantins (UFT), Ciência da Computação, Palmas, Brasil*
[2] *Universidade Federal do Norte do Tocantins (UFNT), Araguaína, Brasil*

Data de recebimento do manuscrito: 03/12/2025
Data de aceitação do manuscrito: 22/01/2026
Data de publicação: 10/02/2026

Resumo—Este artigo apresenta uma abordagem pedagógica para o estudo do problema *Hitting Set*, com o intuito de reproduzir e tornar acessível a demonstração clássica de sua NP-completude, conforme estabelecida na literatura especializada. Diferentemente de trabalhos que visam propor novos resultados teóricos inéditos, o objetivo central desta pesquisa é preencher uma lacuna didática, detalhando minuciosamente a redução polinomial a partir do problema *Vertex Cover* (Problema Alvo) para o *Hitting Set* (Problema Atacado). A metodologia adotada inicia-se com uma revisão dos conceitos fundamentais, incluindo as definições formais das classes P e NP, bem como o conceito de certificado e verificação eficiente. Em seguida, uma prova inspirada na de Richard Karp é construída passo a passo, com ênfase na visualização da transformação das instâncias de grafos para coleções de conjuntos através de diagramas de "antes e depois". Adicionalmente, introduz-se o "Dilema dos Observadores", uma analogia original para ilustrar a complexidade combinatória. Por fim, discutem-se aplicações práticas em bioinformática e engenharia de software, consolidando o material como um recurso de apoio eficaz ao ensino de Teoria da Computação.

Palavras-chave—Hitting Set, NP-Completo, Vertex Cover, Redução Polinomial, Teoria da Computação.

Abstract*—This paper presents a pedagogical approach to the study of the Hitting Set problem, aiming to reproduce and make accessible the classic demonstration of its NP-completeness, as established in the specialized literature. Unlike works aiming to propose novel theoretical results, the central objective of this research is to bridge a didactic gap by meticulously detailing the polynomial reduction from the Vertex Cover problem to the Hitting Set problem. The adopted methodology begins with a review of fundamental concepts, including formal definitions of the P and NP classes, as well as the concepts of certificates and efficient verification. Subsequently, a proof inspired by Richard Karp is constructed step-by-step, emphasizing the visualization of transforming graph instances into set collections using "before and after" diagrams. Additionally, the "Observer's Dilemma" is introduced—an original analogy to illustrate combinatorial complexity. Finally, practical applications in bioinformatics and software engineering are discussed, consolidating the material as an effective support resource for teaching Theory of Computation.*

Keywords*—Hitting Set, NP-Complete, Vertex Cover, Polynomial Reduction, Theory of Computation.*

I. INTRODUÇÃO

O ensino de Teoria da Computação, especificamente no tópico de NP-Completude, impõe desafios significativos aos estudantes de graduação devido ao alto nível de abstração exigido. Compreender formalmente como a dificuldade de um problema pode ser "traduzida" para o conceito de redução polinomial é frequentemente uma barreira de aprendizado que exige mais do que apenas definições matemáticas, exige visualização e intuição. Entre os diversos problemas estudados no âmbito da classe $\mathcal{NP}$, o *Hitting Set* ocupa papel significativo, tanto por sua relevância teórica quanto por sua ampla gama de aplicações práticas conforme discutido por Karp [1] ao apresentar sua formulação clássica no estudo dos problemas NP-Completos. Tendo em vista a união de referências clássicas amplamente adotadas, é perceptível que esses materiais frequentemente

Dados de contato: Benedito Jaime Melo Moraes Junior, benedito.jaime@uft.edu.br

apresentam a prova de NP-completude do Hitting Set de forma condensada, com poucos recursos visuais e saltos lógicos que pressupõem um alto grau de maturidade matemática do leitor. Na prática de sala de aula, observa-se que estudantes de graduação têm dificuldade em acompanhar esses argumentos sem um material intermediário que detalhe a redução passo a passo, com exemplos graduais e analogias concretas. Assim, identifica-se uma lacuna didática entre a literatura de referência, voltada a um público mais avançado, e as necessidades de estudantes em disciplinas introdutórias de Teoria da Computação.

Neste contexto, este trabalho visa oferecer uma reprodução pedagógica da prova de NP-Completude do problema *Hitting Set*. Utilizando a redução clássica a partir do *Vertex Cover* (Cobertura de Vértices) [2], buscamos detalhar as etapas lógicas e fornecer recursos visuais que auxiliem o entendimento da literatura técnica padrão, facilitando a assimilação dos conceitos fundamentais por estudantes iniciantes. Dessa forma, como uma contribuição pedagógica, este trabalho apresenta recursos para facilitar o aprendizado dos conceitos de Teoria da Computação. O material inclui uma prova da NP-Completude do problema Hitting Set, desdobrando os aspectos técnicos para maior clareza. Para tornar os conceitos abstratos mais tangíveis, são fornecidas figuras e exemplos que ilustram tanto o processo de redução polinomial quanto a verificação das soluções. Como parte de uma estratégia lúdica, o estudo incorpora um problema ilustrativo (o "Dilema dos Observadores"), que aproxima o conceito de complexidade combinatória do cotidiano dentro da complexidade explorada na NP-Completude. Por fim, a compreensão da classe $\mathcal{NP}$ é reforçada com a inclusão de pseudocódigo e análise de verificação, demonstrando formalmente a eficiência do algoritmo que checa a validade de uma solução candidata.

A estrutura deste trabalho foi organizada para guiar o leitor desde os fundamentos até a prova formal. Iniciamos revisando os conceitos basilares de grafos e complexidade relacionados ao problema estudado. Em seguida, contextualizamos o problema na literatura, para então definirmos o *Hitting Set* e apresentarmos a demonstração técnica visual, encerrando com uma reflexão sobre as estratégias de aprendizado adotadas, seguindo a metodologia de seminários proposta por Lassance, Bianchini e Santos [3]. O objetivo central deste trabalho, portanto, não é apresentar novos resultados teóricos sobre Hitting Set, mas organizar uma rota de aprendizagem que torne a prova clássica de sua NP-completude acessível a estudantes iniciantes, complementando os livros-texto tradicionais, com ênfase em recursos de visualização que mostrem a transformação das instâncias, demonstrações que gradualmente aproximem o estudante da prova completa e conexões explícitas entre a prova abstrata e aplicações concretas.

II. Preliminares

Para fundamentar a demonstração que será desenvolvida, revisamos nesta seção os conceitos essenciais e estabelecemos a notação utilizada. As definições aqui apresentadas seguem as convenções de Sipser [4] e Garey & Johnson [2].

O primeiro conceito fundamental é o de *grafo*, uma estrutura matemática amplamente utilizada para modelar relações entre objetos, como ligações entre computadores em uma rede, estradas ligando cidades ou conexões entre páginas da web. Formalmente, um grafo é denotado por $G = (V,E)$, onde V representa o conjunto de vértices (ou nós), que são os pontos do grafo, e E representa o conjunto de arestas, que são as conexões entre pares de vértices. Em um grafo simples e não direcionado, cada aresta é um par não ordenado $\{u,v\}$, indicando apenas que existe uma ligação entre u e v, sem sentido de direção. Esse tipo de estrutura é especialmente conveniente para problemas de cobertura, pois permite enxergar relações de conexão de maneira clara.

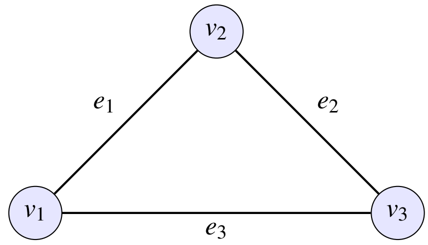

Figura 1: Representação de um grafo não direcionado $G = (V,E)$. As arestas são rotuladas como e_i, e os vértices como v_i.

Para a redução proposta que será apresentada mais adiante, é crucial entender também o conceito de *incidência* e de *grau*. Dizemos que uma aresta $\{u,v\} \in E$ é incidente aos vértices u e v, isto é, ela "toca" exatamente esses dois vértices. O grau de um vértice, por sua vez, é o número de arestas incidentes a ele e indica quantas conexões diretas aquele ponto possui dentro do grafo. A Figura 1 apresenta uma ilustração visual desses componentes: os círculos representam os vértices (V) e as linhas representam as arestas (E). No exemplo, o vértice v_3 possui grau 2, pois está ligado a v_1 e v_2; esse tipo de contagem será reutilizado mais adiante quando mapearemos vértices e arestas para conjuntos e elementos na redução para o problema *Hitting Set*.

Além do conceito de grafos utilizado, é necessário abordar que o contexto deste trabalho exige a definição clara do ambiente de Complexidade Computacional de forma que facilite a compreensão dentro do âmbito conteúdo-aluno.

Finalmente, para realizar a prova de NP-Completude, utilizaremos o conceito de redução e um problema base. O problema escolhido como ponto de partida é o *Vertex Cover*. Sua NP-Completude foi demonstrada por Richard Karp [1], sendo uma das mais aceitas no contexto de cobertura de grafos. Ele é definido pela seguinte instância e questão: dado um grafo $G = (V,E)$ e um inteiro k, é possível escolher um subconjunto de vértices $C \subseteq V$ (com $|C| \leq k$) tal que todas as arestas de E tenham pelo menos uma extremidade em C?

Problema: VERTEX COVER (VC)

Entrada: Um grafo simples $G = (V,E)$ e um inteiro $k \in \mathbb{N}$.

Questão: Existe um subconjunto $C \subseteq V$ com $|C| \leq k$ tal que, para toda aresta $\{u,v\} \in E$, vale $u \in C$ ou $v \in C$? (Ou seja, cada aresta de G possui ao menos uma extremidade em C.)

 ISSN: 2675-3588

Este problema servirá de alicerce para a construção do *Hitting Set* nas seções subsequentes, pois a prova de NP-Completude será obtida por meio de uma redução polinomial de *Vertex Cover* para *Hitting Set*. De maneira geral, o *Hitting Set* recebe como entrada um universo finito de elementos e uma coleção de subconjuntos desse universo, e pergunta se existe um subconjunto H com tamanho limitado por k que intercepte todos esses subconjuntos, isto é, que contenha pelo menos um elemento em comum com cada um deles. Na prática, o *Hitting Set* pode ser visto como uma generalização de problemas de cobertura em grafos, na qual arestas e vértices são substituídos por subconjuntos e elementos de um universo arbitrário.

Problema: HITTING SET (HS)

Entrada: Um conjunto finito U (universo), uma coleção $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ de subconjuntos de U e um inteiro positivo $k \in \mathbb{N}$.

Questão: Existe um subconjunto $H \subseteq U$ com cardinalidade $|H| \leq k$ tal que H intercepte todos os conjuntos de $\mathcal{S}$? (Ou seja, $H \cap S_i \neq \emptyset$ para todo $S_i \in \mathcal{S}$).

III. Trabalhos Relacionados

A fundamentação deste artigo baseia-se em três eixos principais: desenvolvimentos recentes em algoritmos e aplicações para o problema de *Hitting Set*, abordagens contemporâneas para ensino de complexidade computacional e o apoio da literatura basilar para estabelecer relação com as práticas pedagógicas na disciplina de Teoria da Computação. A seguir, destacam-se as obras diretamente relacionadas à proposta.

Do ponto de vista técnico, estudos recentes sobre geração de *Hitting Sets* mínimos e sobre aplicações em biologia de sistemas evidenciam que o *Hitting Set* permanece um problema central tanto na pesquisa teórica quanto em cenários aplicados por Gainer-Dewar, Vera-Licona e Haus[5, 6]. Esses trabalhos discutem algoritmos em contextos reais, reforçando a importância de compreender, mesmo em nível introdutório, por que o problema é intratável e quais estratégias práticas são adotadas na literatura recente.

É importante ressaltar que busca-se a inspiração basilar em trabalhos como o de Garey e Johnson [2], referência em intratabilidade para denotarmos o entendimento em materiais recentes, pois fornecem a definição formal do *Hitting Set* e sua classificação como problema NP-Completo, baseada na equivalência com o *Set Cover*. O trabalho seminal de Karp [1] é utilizado para contextualizar historicamente as reduções polinomiais, técnica central aplicada neste artigo, bem como para situar o *Hitting Set* no panorama dos problemas intratáveis.

No eixo de aplicações, resultados como os de Gainer-Dewar e Vera-Licona [5] e de Haus et al. [6] ilustram o uso de *hitting sets* na análise de redes biológicas e em ambientes de alto desempenho, o que contribui para motivar o estudo do problema junto a estudantes da área de computação e a visão da aplicabilidade no tom pedagógico. Ao mostrar que a mesma estrutura combinatória aparece em contextos atuais de pesquisa, esses trabalhos ajudam a conectar o conteúdo teórico da disciplina com problemas concretos de interesse científico e tecnológico.

Os trabalhos de Chvátal [7], Ammann e Offutt [8] trazem abordagens que alimentam a discussão de aplicabilidade prática. Chvátal discute heurísticas gulosas como forma de contornar a intratabilidade em problemas de cobertura, enquanto Ammann e Offutt conectam a teoria abstrata à prática de testes de software, justificando a relevância do *Hitting Set* para a formação de futuros profissionais.

Do ponto de vista pedagógico, o artigo de Lassance, Bianchini e Santos [3], que descreve o "Ciclo de Seminários em Teoria da Computação", serviu como referência metodológica direta. Dessa experiência, foi adotada a ideia de decompor a prova em "Problema Atacado" (Vertex Cover) e "Problema Alvo" (Hitting Set), bem como a ênfase na construção de recursos visuais e exemplos guiados como suporte à aprendizagem em disciplinas introdutórias. Em complemento, trabalhos que exploram o uso de visualizações, animações e ferramentas interativas para o ensino de NP-Completude, indicam uma tendência recente de tornar as reduções mais acessíveis por meio de abordagens ativas e multimodais, Crescenz e Marchetti [9, 10].

Em conjunto, essas referências não apenas sustentam a prova teórica apresentada nas seções seguintes mas abordam de maneira recente, e fundamentam a escolha de uma abordagem fortemente didática, alinhada com práticas contemporâneas de ensino de complexidade e com aplicações atuais do problema de *Hitting Set*.

IV. Descrição do Problema

O *Hitting Set* é um dos problemas mais dinâmicos na teoria da complexidade, justamente pela sua objetividade que serve como um bom drive de verificação entre problemas tratáveis e intratáveis. Sua classificação como NP-Completo foi estabelecida originalmente por Richard Karp em sua lista seminal de 21 problemas [1], devido à sua equivalência direta com o problema *Set Cover*. Posteriormente, Garey e Johnson [2] consolidaram sua importância como um problema ilustre para provas de redução, dada a sua estrutura combinatória limpa e versátil.

Para compreender a natureza deste problema, é essencial distinguir inicialmente entre as versões de otimização e decisão. Em sua forma natural, o *Hitting Set* é um problema de otimização que busca responder: "Qual é o *menor* número de elementos necessários para atingir todos os subconjuntos?". No entanto, para a classificação na classe $\mathcal{NP}$, utilizamos a versão de decisão, que impõe um limite superior k. A questão central torna-se: "É possível atingir todos os conjuntos utilizando no máximo k elementos?".

Formalmente, seguindo a notação proposta por Garey e Johnson [2], seja U um conjunto finito, chamado de *universo*, e seja $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ uma coleção finita de subconjuntos de U, isto é, $S_i \subseteq U$ para todo $1 \leq i \leq m$. Seja ainda $k \in \mathbb{N}$ um inteiro não negativo. O problema Hitting Set na forma de decisão é definido da seguinte maneira:

TABELA 1: INTRATABILIDADE: COMPARAÇÃO DO NÚMERO DE OPERAÇÕES NECESSÁRIAS CONFORME A ENTRADA n CRESCE.

Entrada (n)	Polinomial (n^2)	Exponencial (2^n)
10	100	1.024
30	900	$\approx$ 1 bilhão
50	2.500	$\approx 10^{15}$
100	10.000	$\approx 10^{30}$

Um subconjunto $H \subseteq U$ que satisfaz $H \cap S_i \neq \emptyset$ para todo $S_i \in \mathcal{S}$ é chamado de *hitting set* (ou *conjunto atingidor*) para a coleção $\mathcal{S}$. Assim, o objetivo do problema é decidir se existe um hitting set de tamanho no máximo k. Nesta notação, U representa o conjunto de todos os elementos disponíveis, $\mathcal{S}$ é a família de subconjuntos que devem ser "atingidos", H é o conjunto solução candidato e k é o limite máximo permitido para o tamanho de H.

A complexidade computacional inerente a esta definição impõe desafios práticos severos. Para encontrar a solução com exatidão, a abordagem mais intuitiva é a chamada Força Bruta. O conceito é simples: o computador testa todas as combinações possíveis de elementos para ver qual é a menor que funciona. É como tentar abrir um cadeado de segredo testando todas as senhas, uma por uma: 000, 001, 002... Embora a Força Bruta seja correta, ela é extremamente lenta por sua natureza combinatória. Dado um universo U com n elementos, o número total de subconjuntos possíveis que podem ser formados é 2^n (incluindo o conjunto vazio). O algoritmo precisa, essencialmente, percorrer todas as 2^n possibilidades, ou pelo menos um grande subconjunto delas, para encontrar a solução ótima. O número de combinações cresce exponencialmente (2^n), tornando a resolução inviável para qualquer instância que não seja muito pequena [2]. Dessa forma, a **Tabela 1** ilustra como esse tempo de execução aumenta rapidamente, por meio da comparação do número de operações necessárias conforme a entrada cresce, baseando-se em análises assintóticas clássicas [4]. A potência 2^n aparece porque, para cada elemento do universo U com n elementos, há duas possibilidades independentes: ou ele entra no subconjunto H ou não entra.

Diante da impossibilidade de verificar todas as opções, pois o número de combinações cresce de forma exponencial, como ilustrado na **Tabela 1**, cientistas da computação recorrem a algoritmos de aproximação [7]. A ideia é aceitar abrir mão da garantia de solução ótima em troca de um algoritmo que rode em tempo polinomial e produza soluções "boas o suficiente" na prática.

De forma explicativa e alinhada à metodologia de ensino proposta por Lassance, Bianchini e Santos [3], adotamos aqui uma estratégia passo a passo voltada ao entendimento dos estudantes. A ideia é construir a solução de forma interativa, sempre observando a instância em um quadro ou diagrama: em cada passo, o aluno identifica quais conjuntos ainda não foram atingidos e escolhe um elemento que ajude a cobrir os casos restantes, atualizando o desenho a cada escolha.

Do ponto de vista algorítmico, essa construção iterativa pode ser vista como uma versão simplificada de uma abordagem gulosa clássica [7]: a cada passo, escolhe-se um elemento que contribui para cobrir muitos conjuntos ainda não atingidos. Esse tipo de estratégia não garante, em geral, a melhor solução possível (ao contrário da Força Bruta, que é exata porém inviável para entradas grandes [2]), mas apresenta duas vantagens fundamentais: (i) seu tempo de execução é polinomial, o que a torna utilizável em instâncias reais, e (ii) existem resultados teóricos que limitam quão pior a solução obtida pode ser em relação à solução ótima [7]. Para fins de análise didática e comparação de crescimento, assumimos aqui um cenário onde o número de subconjuntos m é proporcional ao tamanho do universo n, permitindo expressar a complexidade apenas em função de n. A **Tabela 2** resume esse contraste entre a exatidão da Força Bruta e a praticidade das abordagens iterativas.

TABELA 2: COMPARAÇÃO ENTRE ABORDAGENS PARA O HITTING SET. ($n = |U|$, ASSUMINDO $m \approx n$ PARA SIMPLIFICAÇÃO).

Abordagem	Complexidade de Tempo	Qualidade da Solução
Enumeração Completa (Força Bruta)	$O(2^n)$	Exata (ótima, mas inviável para n grande)
Construção Iterativa Orientada (passo a passo)	$O(n^2)$	Em geral não ótima, mas utilizável na prática

Contudo, em cenários industriais onde a exatidão é inegociável (como no diagnóstico médico ou em configurações de segurança crítica), depender apenas de aproximações pode ser insuficiente. Para esses casos, a indústria recorre à *Complexidade Parametrizada* (FPT - *Fixed-Parameter Tractability*). Nesta abordagem, a complexidade é analisada em função de dois valores: o tamanho da entrada n e um parâmetro fixo k — que, neste problema, corresponde ao tamanho da solução buscada. A estratégia é confinar a "explosão combinatória" exclusivamente a esse parâmetro k, mantendo o tempo polinomial em relação a n. Algoritmos FPT com complexidade do tipo $O(2^k \cdot n)$ exemplificam bem essa vantagem: considere uma base de dados com $n =$ 1.000 elementos onde buscamos um subconjunto de tamanho $k = 10$. Enquanto a abordagem FPT exigiria apenas $\approx 10^6$ operações, sendo resolvida em cerca de 1 milissegundo (supondo 10^9 operações/s), a força bruta (2^n) exigiria $2^{1.000}$ operações, o que levaria um tempo superior à idade do universo para ser concluído. Essa abordagem permite lidar com a intratabilidade de forma cirúrgica em instâncias reais, sem sacrificar a precisão dos resultados [11].

Apesar da dificuldade geral, existem exceções interessantes. Se restringirmos a instância de modo que cada subconjunto em $\mathcal{S}$ tenha tamanho máximo 2 (isto é, $|S_i| \leq 2$ para todo $S_i \in \mathcal{S}$), o problema torna-se idêntico ao *Vertex Cover*, que, apesar de ainda ser NP-Completo, permite análises mais detalhadas e soluções aproximadas bem estudadas em questão literária.

Para visualizar a definição formal na prática, considere a instância apresentada na Figura 2. Neste exemplo, temos o universo $U = \{1,2,3,4,5\}$ e a coleção $\mathcal{S} = \{\{1,2,3\},\{2,4\},\{3,4\},\{4,5\}\}$ com $k = 2$. A resposta é afirmativa, pois o conjunto $H = \{2,4\}$ possui tamanho 2 e intersecta todos os subconjuntos de $\mathcal{S}$. Os elementos da

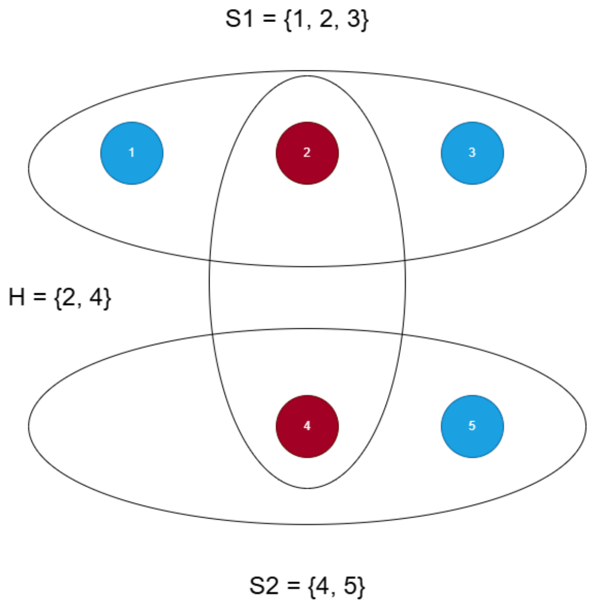

Figura 2: Representação visual de S, no qual $H = \{2,4\}$ é *Hitting Set*.

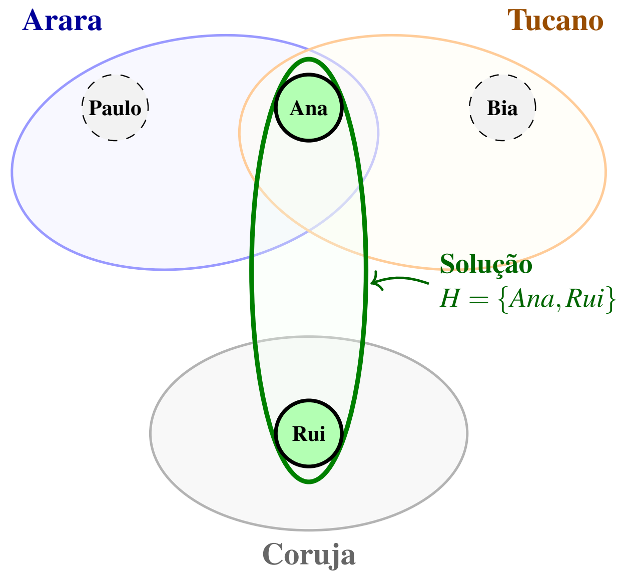

Figura 3: Ana cobre Arara e Tucano; Rui cobre a Coruja. O envelope verde destaca o conjunto solução $H = \{Ana, Rui\}$.

solução $H = \{2,4\}$ estão destacados em vermelho; note que cada elipse (conjunto) contém pelo menos um elemento vermelho.

Apesar da dificuldade geral, existem exceções interessantes. Se restringirmos a instância de modo que cada subconjunto em $\mathcal{S}$ tenha tamanho máximo 2 (isto é, $|S_i| \leq 2$ para todo $S_i \in \mathcal{S}$), o problema torna-se idêntico ao *Vertex Cover*, que, apesar de ainda ser NP-Completo, permite análises mais detalhadas e soluções aproximadas bem estudadas na literatura.

Para visualizar a definição formal na prática, considere a instância apresentada na **Figura 2**. Neste exemplo, temos o universo $U = \{1,2,3,4,5\}$ e a coleção $\mathcal{S} = \{\{1,2,3\},\{4,5\}\}$ com $k = 2$. A resposta é afirmativa, pois o conjunto $H = \{2,4\}$ possui tamanho 2 e intersecta todos os subconjuntos de $\mathcal{S}$. Os elementos da solução $H = \{2,4\}$ estão destacados em vermelho; note que cada elipse (conjunto) contém pelo menos um elemento vermelho.

Para facilitar a intuição sobre a complexidade combinatória, propomos uma analogia original denominada "O Dilema dos Observadores". A ideia é traduzir a definição formal do *Hitting Set* para uma narrativa concreta, em que os elementos do universo e os subconjuntos ganham interpretação no mundo real. Essa analogia é inspirada no clássico problema de *Crew Scheduling* (Escalonamento de Tripulações), citado por Garey e Johnson [2] como uma aplicação canônica de problemas de cobertura de conjuntos.

> "Uma equipe de biólogos precisa confirmar a presença de 5 espécies raras de pássaros (S_1 a S_5) em uma reserva. Eles têm 10 observadores disponíveis. Cada observador é especialista em identificar um subconjunto diferente de espécies. A equipe tem orçamento para contratar no máximo k observadores. A pergunta é: *é possível formar um time com $\leq k$ pessoas que identifique **todas** as espécies?*"

Nessa analogia, o conjunto de observadores é o universo U e cada espécie define um subconjunto S_i de quem pode avistá-la. Um conjunto de observadores contratados é um *Hitting Set*. A **Figura 3** ilustra uma instância desse dilema.

A personagem Ana (em vermelho) é uma generalista que cobre duas espécies (Arara e Tucano). Porém, ao escolhê-la, ainda precisamos cobrir a Coruja, que só é vista pelo Rui. Assim, uma solução possível seria o time $\{Ana, Rui\}$ (tamanho 2). Outra opção seria ignorar a Ana e contratar apenas especialistas dedicados: $\{Paulo, Bia, Rui\}$ (tamanho 3).

O "dilema" computacional é que não existe uma regra simples (como "sempre escolha quem cobre mais") que garanta a melhor solução em todos os casos. O computador precisa verificar as diversas combinações (Generalistas vs. Especialistas) para garantir que o orçamento k seja respeitado.

Além do interesse teórico, o *Hitting Set* modela desafios reais onde a eficiência é crítica. Na bioinformática, é aplicado na seleção de marcadores genéticos [1]. Na engenharia de software, é utilizado para minimizar suítes de teste [8]. Como o problema é NP-Completo, a inviabilidade da força bruta valida o uso das heurísticas de aproximação como a abordagem padrão na indústria.

Essa analogia faz sentido em relação ao problema de *Hitting Set* porque traduz, quase literalmente, cada componente da definição formal para elementos intuitivos da história, permitindo ao estudante "ver" o problema em vez de apenas manipulá-lo simbolicamente. Do ponto de vista matemático, o universo U do *Hitting Set* corresponde ao conjunto de observadores disponíveis, enquanto cada subconjunto $S_i \in \mathcal{S}$ é interpretado como o grupo de observadores capazes de identificar a espécie i. Um *hitting set* $H \subseteq U$ é um conjunto com elementos que intersecta todos os S_i; na analogia, isso significa escolher um time de observadores tal que, para cada espécie rara, pelo menos um membro do time consiga identificá-la. O parâmetro k que limita o tamanho de H é modelado diretamente pelo orçamento máximo de observadores que podem ser contratados.

ISSN: 2675-3588

Além de respeitar essa correspondência estrutural, o dilema também ajuda a construir intuição sobre a complexidade do problema. Por um lado, evidencia o caráter combinatório: há muitas formas possíveis de escolher subconjuntos de observadores, e nem todas cobrem todas as espécies, o que espelha o grande espaço de soluções candidatas no *Hitting Set*. Por outro lado, ilustra o perigo de decisões puramente locais: escolher a observadora "generalista" Ana parece uma boa escolha quando se olha apenas para o número de espécies cobertas, mas não resolve o caso da Coruja, exigindo a presença do especialista Rui. Dessa forma, a narrativa mostra que a melhor decisão local nem sempre leva à melhor solução global, um ponto central em problemas NP-Difíceis.

É imperativo, contudo, delimitar o escopo desta analogia lúdica para evitar simplificações excessivas. O "Dilema dos Observadores" atua estritamente como um facilitador para a compreensão do *enunciado* e das restrições do problema, não substituindo a formalização matemática necessária para a análise de complexidade. Em cenários cotidianos ou administrativos, como o descrito na narrativa, a intuição humana frequentemente encontra padrões que facilitam a resolução. No entanto, a classificação de NP-Completude lida com instâncias arbitrárias de "pior caso", onde tais padrões intuitivos inexistem ou são enganosos. Portanto, a analogia serve como porta de entrada cognitiva, mas o rigor algébrico — detalhado na demonstração da Seção V — permanece insubstituível para a validação científica da intratabilidade.

Em síntese, o "Dilema dos Observadores" funciona como um modelo mental que o aluno pode reutilizar nas seções seguintes: sempre que se deparar com a notação U, $\mathcal{S}$, H e k, pode lembrar de observadores, espécies e orçamento, o que reduz a carga cognitiva e facilita a compreensão das provas formais.

V. Demonstração e Contribuições Técnicas

Esta seção apresenta a sistematização da prova de NP-Completude do *Hitting Set*. Diferentemente dos manuais técnicos que priorizam a concisão, optamos aqui por uma abordagem expandida, detalhando os passos lógicos que frequentemente são omitidos na literatura especializada [2].

Para classificar formalmente um problema como NP-Completo, é necessário satisfazer simultaneamente duas condições: provar que $HS \in \mathcal{NP}$, e que $HS \in$ NP-Difícil. Essas provas serão feitas a seguir.

Lema 1. *O problema HS pertence a $\mathcal{NP}$.*

Proof. Seguindo esse fluxo lógico, o primeiro passo é demonstrar que o problema pertence à classe $\mathcal{NP}$. Isso exige a existência de um algoritmo que, dada uma solução candidata (certificado), consiga verificar sua validade em tempo eficiente, conforme a definição formal de verificadores polinomiais estabelecida por Sipser [4]. No contexto do *Hitting Set*, considere uma instância definida por um universo de elementos U, uma coleção S de subconjuntos de U e um inteiro k [2]. O certificado é um subconjunto candidato $H \subseteq U$. O algoritmo verificador recebe a instância (U,S,k) e o certificado H, respondendo "Sim" apenas se duas condições forem satisfeitas: (1) o tamanho de H respeita o limite k (i.e., $|H| \leq k$); e (2) H intersecta todos os subconjuntos de S. Abaixo apresentamos o algoritmo que realiza essa verificação:

```
Algoritmo Verificador(U, S, H, k)
Início
   Se (tamanho(H) > k) então
      Retorne Falso;
   Fim-Se
   Para cada S_i em S faça
      Verifica se a interseção é vazia
      Se (H ∩ S_i == ∅) então
         Retorne Falso;
      Fim-Se
   Fim-Para
   Retorne Verdadeiro;
Fim
```

Para realizar a análise de eficiência deste algoritmo (tecnicamente chamada de análise assintótica [4]), definimos n como o tamanho total da entrada recebida pelo algoritmo (a soma dos tamanhos de U, S, H e a representação de k). É fácil analisar que a verificação de tamanho é uma operação linear $O(n)$, pois, no pior caso, o algoritmo precisa percorrer a lista de elementos de H para contá-los, e o tamanho de H nunca excede o tamanho total da entrada n. Já a Verificação de Cobertura é a etapa dominante: sua estrutura de repetição obriga a comparação dos elementos de H com os de cada subconjunto em S, resultando em uma complexidade quadrática $O(n^2)$. Como n^2 é um polinômio, garantimos que a verificação é eficiente.

Desta forma, no pior cenário possível, essa estrutura faz o algoritmo comparar sistematicamente os elementos, resultando em um número total de operações proporcional ao produto $m \times n$ (onde n é o tamanho do universo). Em termos de complexidade, isso é representado pela notação $O(n^2)$, indicando que o tempo cresce quadraticamente em relação ao tamanho da entrada [4]. Como uma função quadrática é um polinômio (e não uma exponencial como 2^n), garantimos que a verificação é computacionalmente viável, confirmando assim que o *Hitting Set* pertence à classe $\mathcal{NP}$ [2]. □

Uma vez estabelecida a NP-Pertinência, o próximo passo do núcleo da prova reside na demonstração de NP-Dificuldade. A estratégia utilizada é a redução polinomial, onde transformamos instâncias de um problema conhecido como NP-Difícil (**Problema Alvo**) em instâncias do problema que queremos classificar (**Problema Atacado**). Para este artigo, reduziremos o *Vertex Cover* ao *Hitting Set* ($VC \leq_p HS$).

Lema 2. *O problema HS é NP-Difícil.*

Proof. Com o problema de partida formalmente definido, passamos à construção da redução. O objetivo desta etapa é demonstrar um algoritmo que transforme, em tempo polinomial, qualquer instância de *Vertex Cover* em uma instância equivalente de *Hitting Set* [1]. Essa transformação deve garantir que a estrutura topológica do grafo seja preservada na forma de conjuntos, de modo que a existência de uma solução em um problema implique diretamente a existência de solução no outro [4].

 ISSN: 2675-3588

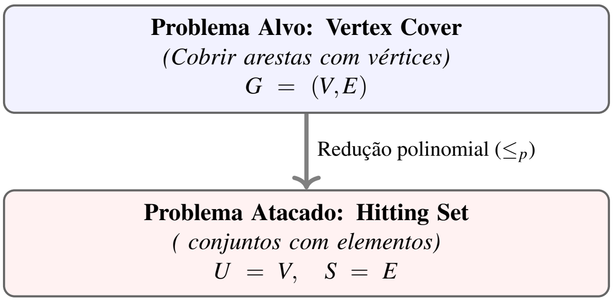

Figura 4: Esquema da redução: transformamos a estrutura do grafo (Atacado) em uma estrutura de conjuntos (Alvo).

Seguindo este raciocínio, no *Vertex Cover* devemos garantir que cada aresta seja coberta por um vértice. No *Hitting Set*, a obrigação é garantir que cada subconjunto seja interceptado por um elemento. Portanto, a estratégia consiste em converter cada aresta (que conecta dois vértices) em um subconjunto (contendo dois elementos) [2]. O esquema conceitual dessa estratégia é apresentado na **Figura 4**.

É fundamental observar que a redução proposta realiza uma tradução da estrutura topológica do grafo para uma estrutura combinatória de conjuntos. Neste contexto, a estrutura combinatória refere-se à organização de elementos baseada estritamente em relações de pertinência e agrupamento, abstraindo qualquer noção de conectividade espacial ou adjacência visual típica dos grafos. A relação de adjacência entre vértices, representada pelas arestas, é remapeada para uma relação de inclusão em subconjuntos.

Desta maneira, a restrição topológica de "cobrir uma aresta" (garantir que uma conexão seja vigiada) é reformulada como a necessidade algébrica de "interceptar um subconjunto" (garantir que um grupo contenha um elemento selecionado). O mapeamento é definido da seguinte maneira: o universo U é constituído pelos vértices de V; a coleção S é formada convertendo cada aresta $\{u, v\}$ em um subconjunto contendo exatamente esses vértices; e o parâmetro de otimização k' preserva seu valor original ($k' = k$). A **Figura 5** ilustra essa transformação por meio da conversão da aresta $\{1, 2\}$ no subconjunto $S_i = \{1, 2\}$.

A corretude desta redução depende da prova de que a instância construída preserva a resposta da original. Demonstramos isso através de duas proposições:

Proposição 1 (Ida $\Rightarrow$): Se G possui um Vertex Cover de tamanho k, então S possui um Hitting Set de tamanho k.

Seja $C \subseteq V$ o Vertex Cover. Escolhemos $H = C$. Para qualquer conjunto $S_i \in S$, sabemos pela construção que ele corresponde a uma aresta $\{u, v\} \in E$. Como C cobre todas as arestas, ele deve conter u ou v. Logo, H contém u ou v, interceptando S_i. Portanto, H é um Hitting Set válido.

Proposição 2 (Volta $\Leftarrow$): Se S possui um Hitting Set de tamanho k, então G possui um Vertex Cover de tamanho k.

Seja $H \subseteq U$ o Hitting Set. Escolhemos $C = H$. Para qualquer aresta $e = \{u, v\} \in E$, existe um conjunto correspondente $S_e = \{u, v\}$ em S. Como H atinge todos os conjuntos, ele deve conter u ou v. Logo, C contém uma extremidade da aresta e. Portanto, C cobre todas as arestas de G.

A Figura 6 ilustra a equivalência lógica. No caso mostrado, a aresta $\{1, 2\}$ é coberta no *Vertex Cover* pelo vértice 1 (destacado em vermelho). Na construção do *Hitting Set*, o conjunto correspondente $S_e = \{1, 2\}$ é atingido pelo mesmo elemento 1, preservando a equivalência entre as duas estruturas.

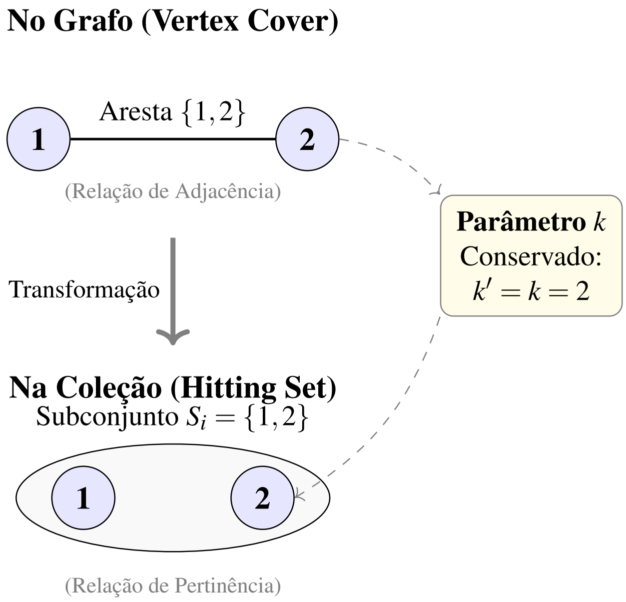

Figura 5: Visualização da Construção: A aresta conectando 1 e 2 no grafo é convertida em um conjunto $S_i = \{1, 2\}$.

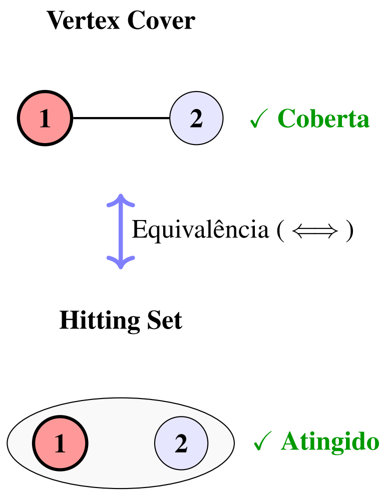

Figura 6: Cobrir a aresta $\{1, 2\}$ com o vértice 1 (vermelho) corresponde a atingir o conjunto $\{1, 2\}$ com o elemento 1.

Desta forma, a partir das provas demonstradas da **Proposição 1** e **Proposição 2**, podemos concluir que HS $\in$ NP-Difícil.

□

Lema 3. *O problema HS é NP-Completo.*

Proof. Para demonstrar que um problema é NP-Completo, é preciso demonstrar que ele pertença simultaneamente as classes $\mathcal{NP}$ e NP-Difícil. Essas demonstrações foram feitas e provadas respectivamente no **Lema 1** e **Lema 2**.

Desta forma, podemos concluir que o problema *Hitting Set* é NP-Completo.

□

Como contribuição pedagógica final, é importante alertar sobre uma armadilha comum no estudo de reduções: a direção da prova. Estudantes frequentemente tentam reduzir

ISSN: 2675-3588

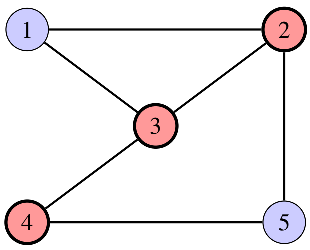

Figura 7: ANTES (Vertex Cover): O grafo de entrada com $k = 3$. Os vértices em vermelho $\{2,3,4\}$ cobrem todas as arestas.

Universo: $U = \{1,2,3,4,5\}$

Coleção S (baseada nas arestas):
$\{1,2\},\{1,3\},\{2,3\},$
$\{2,5\},\{3,4\},\{4,5\}$

Solução Mapeada:
$H = \{2,3,4\}$

Figura 8: DEPOIS (Hitting Set): A instância resultante. O conjunto $H = \{2,3,4\}$ intercepta todos os subconjuntos listados.

o problema novo para o problema conhecido ($HS \leq_p VC$). Isso provaria apenas que o HS é "fácil" o suficiente para ser resolvido pelo VC, mas não que ele é "difícil". A prova de NP-Dificuldade exige o oposto: mostrar que o problema novo é capaz de simular qualquer instância do problema difícil conhecido ($VC \leq_p HS$).

VI. Resultados e Reflexões

A elaboração deste artigo permitiu consolidar o entendimento sobre a hierarquia de complexidade e as técnicas de redução polinomial. Mais do que a demonstração formal, o principal produto deste trabalho é a sistematização didática apresentada, que busca preencher lacunas de compreensão comuns em estudantes iniciantes. A visualização do mapeamento entre instâncias mostrou-se uma ferramenta poderosa para tangibilizar a abstração algébrica.

Uma reflexão crítica sobre a metodologia adotada revela que a escolha do *Vertex Cover* como problema de partida (Problema Atacado) foi determinante para a clareza da exposição. Embora a literatura clássica frequentemente utilize reduções a partir de problemas lógicos como o 3-SAT, essa abordagem exige que o estudante transite entre o domínio da lógica booleana e a teoria dos conjuntos, o que adiciona uma carga cognitiva extra. Ao optarmos por uma redução grafo-para-conjunto ($VC \leq_p HS$), mantivemos a natureza visual do problema, permitindo que a transformação seja verificada "a olho nu", como ilustrado na sequência da **Figura 7** e **Figura 8**.

A construção dessas contribuições pedagógicas foi o foco central. Em vez de presumir conhecimento prévio, dedicamos as seções iniciais a explicar termos essenciais utilizando analogias. Um dos pontos altos foi o uso do "Dilema dos Observadores" para explicar os fundamentos teóricos: utilizamos essa analogia para concretizar que verificar uma solução (conferir a equipe contratada) é rápido, mas encontrar a solução ótima (testar todas as combinações) é exponencialmente difícil. Essa distinção é crucial para que o estudante compreenda a natureza da classe $\mathcal{NP}$ não como uma medida de "impossibilidade", mas como uma medida de "custo de busca".

Ainda sobre a estratégia lúdica, é pertinente observar que o "Dilema dos Observadores" também serve para ilustrar as limitações das abordagens intuitivas. Em sala de aula, é comum que alunos sugiram algoritmos gulosos (como contratar sempre a pessoa mais versátil) como solução geral. A modelagem do problema permitiu demonstrar que, em cenários de complexidade NP-Completa, a intuição local falha diante da necessidade de uma otimização global, validando a necessidade de rigor matemático na análise de algoritmos.

No entanto, o processo de elaboração deste material não foi isento de dificuldades. O maior desafio encontrado não foi a complexidade técnica da prova em si — pois a redução de *Vertex Cover* é direta — mas sim o desafio de transposição didática: explicar os fundamentos sem recorrer a jargões herméticos que afastam o leitor iniciante. A estratégia adotada de explicar cada conceito técnico (como a análise assintótica) imediatamente antes de sua aplicação mostrou-se essencial para manter a clareza e a acessibilidade do texto.

Quanto à aplicabilidade acadêmica, este trabalho foi feito para servir como um material complementar para futuros alunos da disciplina de Teoria da Computação. Acreditamos que a exposição visual da redução e a discussão sobre as nuances entre decisão e otimização oferecem um ponto de entrada mais suave para o tema. Tanto este artigo quanto as referências discutidas podem ser usados como um guia introdutório e acessível para quem precisa entender como uma prova de NP-Completude é estruturada na prática, cumprindo o objetivo de facilitar o aprendizado e desmistificar a teoria.

VII. Conclusão

A elaboração deste estudo permitiu atingir o objetivo principal de demonstrar a NP-Completude do problema *Hitting Set* de forma pedagógica seguindo o rigor exigido pela literatura clássica. A prova foi estruturada em duas etapas fundamentais: a verificação de pertinência à classe $\mathcal{NP}$, realizada através da análise de um algoritmo verificador polinomial, e a demonstração de NP-Dificuldade, executada por meio da redução canônica a partir do *Vertex Cover*. Este resultado teórico não é apenas um rótulo classificatório; ele carrega uma implicação prática profunda: a menos que $\mathcal{P} = \mathcal{NP}$, não existem algoritmos eficientes para resolver o *Hitting Set* de forma exata em todos os casos, validando a necessidade de abordagens aproximadas.

Do ponto de vista pedagógico, o material foi aplicado em uma turma de Teoria da Computação, envolvendo aproximadamente 20 estudantes de graduação.[1] Em uma atividade de seminário, os alunos foram convidados a reconstruir a redução $VC \leq_p HS$ utilizando os diagramas apresentados e a reprodução guiada das etapas da prova, antes do contato direto com os livros-texto formais. Nessa dinâmica, as interações e discussões em sala facilitaram o compartilhamento de diferentes formas de explicar a redução, em uma linguagem mais próxima dos próprios estudantes, mediadas pela equipe de pesquisadores. Observou-se que os alunos passaram a demonstrar maior segurança para

[1] Relato de aplicação didática conforme descrito na seção de Resultados e Reflexões.

 ISSN: 2675-3588

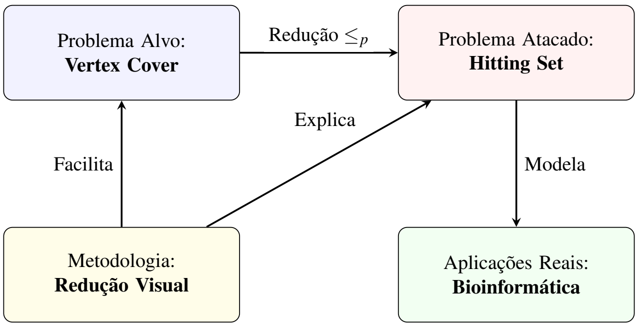

Figura 9: Mapa síntese da abordagem: A metodologia visual conecta o problema base ao alvo.

explicar, com suas próprias palavras, o papel do certificado em $\mathcal{NP}$ e o encadeamento lógico da redução, apoiados por estratégias visuais organizadas em slides interativos, com tempo de exposição limitado para evitar sobrecarga cognitiva. Ainda que esses registros não constituam um estudo quantitativo rigoroso, eles fornecem indícios qualitativos de que a abordagem visual e narrativa contribuiu para reduzir a sensação de "salto conceitual" frequentemente associada às provas de NP-completude [9, 10].

Para além da demonstração matemática, o contributo mais expressivo deste trabalho reside na sua proposta pedagógica. Conforme as diretrizes da disciplina, buscou-se transpor a barreira da abstração que frequentemente dificulta o aprendizado de Teoria da Computação. A introdução do "Dilema dos Observadores" serviu como uma ponte cognitiva, traduzindo a aridez da notação de conjuntos para um problema tangível de gestão de recursos. Essa analogia facilitou a intuição sobre a assimetria fundamental da complexidade: a facilidade de verificar uma solução dada (auditar uma equipe contratada) em contraste com a dificuldade de encontrar a solução ótima (testar todas as combinações de equipes).

Para consolidar a jornada de aprendizado proposta, a **Figura 9** apresenta um mapa conceitual que resume a estrutura lógica desenvolvida no artigo, conectando a teoria de base, a prova de redução e as aplicações práticas.

Embora a sistematização proposta tenha êxito em seus objetivos didáticos, o trabalho apresenta limitações no seu escopo, concentrando-se majoritariamente no aspecto teórico da classificação de complexidade. Não foram exploradas, nesta etapa, implementações computacionais de algoritmos de aproximação como a Heurística Gulosa [7] ou algoritmos parametrizados (FPT), que constituem a abordagem padrão para lidar com a intratabilidade do problema em cenários industriais reais [8, 1]. Adicionalmente, a redução restringiu-se ao caminho clássico via *Vertex Cover*, sem explorar reduções alternativas que poderiam oferecer outras perspectivas.

Como desdobramento natural deste estudo, trabalhos futuros podem focar na vertente experimental, implementando e comparando o desempenho de algoritmos exatos (para instâncias pequenas) versus algoritmos aproximativos (para instâncias grandes). Outra via promissora seria o aprofundamento em classes especiais de instâncias, como aquelas com restrições de cardinalidade nos subconjuntos, investigando cenários onde o problema se torna tratável e enriquecendo ainda mais o repertório de exemplos didáticos disponíveis para o ensino de Computação.

REFERÊNCIAS

[1] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*. Springer, 1972, pp. 85–103.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[3] Y. M. Lassance, G. d. B. Bianchini, and T. D. Santos, "Reflexões e práticas pedagógicas no escopo da disciplina de teoria da computação," *Academic Journal on Computing, Engineering and Applied Mathematics (AJCEAM)*, vol. 6, no. 2, pp. 10–17, 2025.

[4] M. Sipser, *Introduction to the Theory of Computation*, 3rd ed. Cengage Learning, 2012.

[5] A. Gainer-Dewar and P. Vera-Licona, "The minimal hitting set generation problem: Algorithms and computation," *SIAM Review*, vol. 58, no. 3, pp. 448–491, 2016.

[6] U.-U. Haus *et al.*, "Finding exact hitting set solutions for systems biology applications using heterogeneous gpu clusters," *Future Generation Computer Systems*, vol. 67, pp. 418–429, 2016.

[7] V. Chvátal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.

[8] P. Ammann and J. Offutt, *Introduction to Software Testing*. Cambridge University Press, 2016.

[9] P. Crescenzi, "Using avs to explain np-completeness," in *Proceedings of the 5th International Conference on Theory and Applications of Diagrams*. Springer, 2010, pp. 95–111.

[10] K. Marchetti *et al.*, "Redux: An interactive, dynamic knowledge base for teaching np-completeness," in *Proceedings of the 2024 ACM Conference on Innovation and Technology in Computer Science Education*. ACM, 2024, pp. 255–261.

[11] R. G. Downey and M. R. Fellows, *Fundamentals of Parameterized Complexity*, ser. Texts in Computer Science. Springer, 2013.

ISSN: 2675-3588