REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

APLICAÇÕES DA GEOMETRIA DO HIPERCUBO NA TEORIA DOS GRAFOS USANDO PYTHON

APPLICATIONS OF HYPERCUBE GEOMETRY IN GRAPH THEORY USING PYTHON

APLICACIONES DE LA GEOMETRÍA DEL HIPERCUBO EN LA TEORÍA DE GRAFOS USANDO PYTHON

Ivam Galvão Filho[1], Vania Ribas Ulbricht[2]

e747703
https://doi.org/10.47820/recima21.v7i4.7703

PUBLICADO: 04.2026

RESUMO

Este artigo trata a geometria do hipercubo e suas aplicações na teoria dos grafos e em problemas computacionais, destacando suas propriedades estruturais e seu potencial de uso em contextos de análise e modelagem. O objetivo central do estudo é identificar as soluções computacionais que utilizam a arquitetura do hipercubo e o grafo hipercubo na resolução de diferentes tipos de problemas. Busca-se compreender como essas estruturas são empregadas em distintos contextos, evidenciando suas características, vantagens e possibilidades de aplicação em diversas áreas da computação. A abordagem metodológica possui caráter teórico e dedutivo, apoiando-se em definições, além do uso de ilustrações gráficas e simulações computacionais para apoiar a compreensão das estruturas analisadas. Para a condução da pesquisa, adotou-se o método de revisão integrativa da literatura, orientado pelo protocolo PRISMA, com a realização de buscas sistemáticas nas bases SciELO, IEEE Xplore e MyEBSCO. Ao todo, foram analisados 11 artigos selecionados a partir das buscas realizadas, além de 8 trabalhos previamente incluídos devido à sua relevância teórica para o tema, totalizando 19 estudos examinados. Como resultado complementar da revisão, foi desenvolvida uma biblioteca em linguagem Python contendo implementações computacionais identificadas na literatura, com o objetivo de apoiar a exploração prática das soluções encontradas.

PALAVRAS-CHAVE: Hipercubo. Teoria dos Grafos. Grafo Hipercubo.

ABSTRACT

This article discusses hypercube geometry and its applications in graph theory and computational problems, highlighting its structural properties and potential use in analysis and modeling contexts. The central objective of the study is to identify computational solutions that utilize the hypercube architecture and hypercube graph in solving different types of problems. It seeks to understand how these structures are employed in different contexts, highlighting their characteristics, advantages, and application possibilities in various areas of computing. The methodological approach is theoretical and deductive, relying on definitions, as well as the use of graphical illustrations and computational simulations to support the understanding of the analyzed structures. For the research, the integrative literature review method was adopted, guided by the PRISMA protocol, with systematic searches in the SciELO, IEEE Xplore, and MyEBSCO databases. In total, 11 articles selected from the searches were analyzed, in addition to 8 works previously included due to their theoretical relevance to the topic, totaling 19 studies examined. As a complementary result of the

[1] Doutorando no Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento - PPGEGC – Universidade Federal de Santa Catarina (UFSC) Florianópolis, Santa Catarina – Brasil.
[2] Doutora, Professora no Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento - PPGEGC – Universidade Federal de Santa Catarina (UFSC) Florianópolis, Santa Catarina – Brasil.

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

review, a Python library was developed containing computational implementations identified in the literature, with the aim of supporting the practical exploration of the solutions found.

KEYWORDS: *Hypercube. Graph Theory. Hypercube Graph.*

RESUMEN
Este artículo analiza la geometría del hipercubo y sus aplicaciones en la teoría de grafos y problemas computacionales, destacando sus propiedades estructurales y su potencial uso en contextos de análisis y modelado. El objetivo central del estudio es identificar soluciones computacionales que utilicen la arquitectura y el grafo del hipercubo para resolver diferentes tipos de problemas. Busca comprender cómo se emplean estas estructuras en distintos contextos, resaltando sus características, ventajas y posibilidades de aplicación en diversas áreas de la computación. El enfoque metodológico es teórico y deductivo, basado en definiciones, así como en el uso de ilustraciones gráficas y simulaciones computacionales para facilitar la comprensión de las estructuras analizadas. Para la investigación, se adoptó el método de revisión bibliográfica integradora, guiado por el protocolo PRISMA, con búsquedas sistemáticas en las bases de datos SciELO, IEEE Xplore y MyEBSCO. En total, se analizaron 11 artículos seleccionados de las búsquedas, además de 8 trabajos previamente incluidos por su relevancia teórica para el tema, lo que suma un total de 19 estudios examinados. Como resultado complementario de la revisión, se desarrolló una biblioteca de Python que contiene implementaciones computacionales identificadas en la literatura, con el objetivo de apoyar la exploración práctica de las soluciones encontradas.

PALABRAS-CLAVE*: Hipercubo. Teoría de grafos. Grafo hipercubo.*

1. INTRODUÇÃO

O hipercubo é um objeto geométrico de grande relevância na matemática discreta, caracterizado como a generalização do quadrado e do cubo para dimensões superiores. Sua estrutura apresenta propriedades notáveis, como simetria, regularidade, diâmetro limitado e alta conectividade, que o tornam adequado para modelagem de sistemas complexos. Quando representado como grafo, denominado grafo de hipercubo, cada vértice corresponde a uma sequência binária de comprimento n, e cada aresta conecta dois vértices cujas sequências diferem em exatamente uma única coordenada (ou seja, em apenas um *bit*). Essa representação permite explorar o hipercubo em contextos computacionais, oferecendo vantagens em termos de otimização de algoritmos, paralelismo e roteamento de redes. Além disso, os grafos de hipercubo apresentam potencial de aplicação em problemas de análise de redes, onde suas propriedades estruturais podem facilitar a detecção de padrões, cálculo de métricas de centralidade, avaliação de robustez e compreensão de relações complexas entre elementos de uma rede. Observa-se, ainda, a escassez de exemplos de aplicações computacionais práticas envolvendo o hipercubo desenvolvidas em linguagem Python. Embora existam diversas aplicações científicas implementadas em múltiplas linguagens de programação, muitas delas são pouco difundidas ou adotadas, o que dificulta sua reutilização e replicabilidade. O estudo do hipercubo e de seus grafos também não se restringe apenas ao campo da teoria matemática, mas se estende a aplicações práticas, estabelecendo uma ponte entre conceitos abstratos e ferramentas computacionais. Dessa

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

forma, essas estruturas podem ser exploradas em diferentes contextos, contribuindo para o desenvolvimento de soluções em diversas áreas da ciência e da engenharia. Este trabalho busca, portanto, investigar as propriedades geométricas e combinatórias do hipercubo, analisar suas representações como grafos e explorar suas aplicações potenciais em diversos cenários. Para atingir este objetivo colocou-se como questão de pesquisa: Como a geometria do hipercubo e as propriedades dos grafos de hipercubo podem ser exploradas em aplicações computacionais? Para responder a esta pergunta de pesquisa colocou-se como objetivo principal: Investigar como a geometria do hipercubo e as propriedades dos grafos de hipercubo podem ser aplicadas em contextos computacionais, com foco em simulações de grafos multidimensionais.

A investigação sobre o hipercubo e grafos apresentam relevância significativa tanto no âmbito teórico quanto aplicado. Do ponto de vista matemático, o hipercubo constitui um objeto geométrico de alta dimensão com propriedades estruturais notáveis, como regularidade, simetria, conectividade e diâmetro limitado, que são fundamentais para a compreensão de grafos complexos e a modelagem de estruturas combinatórias. Do ponto de vista computacional, os grafos de hipercubo oferecem vantagens importantes em aplicações que envolvem algoritmos paralelos, redes de comunicação e análise de grandes volumes de dados, devido à sua topologia altamente conectada e previsível. Além disso, a aplicação desses grafos em análise de redes complexas representa uma oportunidade para explorar novas abordagens em contextos práticos, como redes sociais, redes de informação e sistemas distribuídos. A pesquisa justifica-se pela necessidade de compreender e explorar de maneira sistemática as propriedades do hipercubo, seus grafos e suas aplicações computacionais, oferecendo contribuições tanto para a matemática discreta quanto para a ciência da computação. Ademais, os resultados esperados podem servir como base para o desenvolvimento de ferramentas e *frameworks* que auxiliem na modelagem e análise de redes complexas, consolidando o hipercubo como uma alternativa prática e teórica relevante em estudos futuros. Diante do exposto, torna-se evidente a necessidade de compreender, de forma sistemática, as propriedades matemáticas e estruturais do hipercubo e de seus grafos, de modo a fundamentar suas aplicações em computação.

De acordo com Shyama e Iyer (2023), o hipercubo apresenta propriedades únicas que o tornam uma estrutura de rede especialmente atrativa para aplicações em computação paralela. Nessa mesma direção, Sundara Rajan *et al.* (2020) destacam que o hipercubo reúne características que o colocam entre as principais opções de topologia para sistemas de processamento e computação paralelos. Outra característica relevante é a capacidade de imersão, que permite que algoritmos originalmente desenvolvidos para uma determinada arquitetura possam ser mapeados em outra. Essa propriedade, chamada de *embeddings* é especialmente importante para a computação paralela, pois confere ao hipercubo a habilidade de simular arquiteturas paralelas de forma eficiente (TAKEMOTO, 1998). Neste artigo, a próxima seção apresenta a revisão de literatura e a fundamentação teórica, na qual serão abordados os conceitos essenciais da geometria do

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

hipercubo, a definição formal de grafos de hipercubo e suas principais propriedades estruturais, além de noções fundamentais da teoria dos grafos e apresentação de aplicações computacionais. Essa base teórica é indispensável para que se possa explorar de maneira consistente as aplicações práticas do hipercubo, permitindo a formulação de hipóteses, a realização de experimentos exploratórios e a proposição de estratégias de utilização em problemas computacionais.

2. REFERENCIAL TEÓRICO

Na revisão de literatura optou-se pela implementação da revisão integrativa com consultas às bases de dados SciElo, uma das mais importantes bases de periódicos nacional, da IEEE Xplore e MyEBSCO, reconhecidas mundialmente pela excelência em publicações de natureza técnico-científica. Para este fim, optou-se pela seguinte *string* de consulta, (*graph*) AND (*hypercube*) e (grafo) AND (hipercubo), aplicada somente a artigos que possuíam os termos no título e resumos, além dos seguintes critérios de inclusão e exclusão apresentados no Quadro 1.

Quadro 1. Critérios de inclusão e exclusão

Critérios de Inclusão		
Nº	**Critério**	**Descrição**
1	Idiomas	Português e inglês
2	Periódicos	Foram aceitos somente artigos publicados em jornais e revistas científicas e de acesso público ou com autenticação por instituição de ensino superior
3	Período	Foram aceitos somente artigos no período de 2016 a 2025
4	Tema	Relacionados a hipercubos, grafos de hipercubos e suas aplicações
Critérios de Exclusão		
Nº	**Critério**	**Descrição**
1	Tema	Foram excluídos textos que não abordam o tema de pesquisa, além de artigos que não apresentam soluções computacionais
2	Período	Não foram aceitos artigos publicados de forma antecipada

Fonte: Dos autores

Também foram integrados na revisão de forma prévia, textos considerados relevantes ao estudo, como livros, páginas *web* e artigos entre outros tipos de documentos, que tratam do tema, mesmo que de forma isolada, sem critério de período.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

2.1. Resultados da Revisão de Literatura

A revisão de literatura resultou em 52 artigos consultados, sendo 2 da SciElo, 14 da IEEE Xplore e 36 da MyEBSCO, na primeira fase da revisão. Nesta fase não houve resultados para buscas em português, somente para a *string* em inglês. Nas plataformas IEEE Xplore e MyEBSCO foram utilizadas credenciais acadêmicas para autenticação e busca. Destes artigos, 8 artigos encontrados na IEEE Xplore foram excluídos por falta de acesso, ou ainda por estarem de acordo com critérios relacionados ao tema da pesquisa. Também foi excluído da pesquisa 1 artigo da SciElo por não se adequar ao tema de pesquisa. Dos 36 artigos encontrados na MyEBSCO, 32 foram excluídos por não estarem de acordo com o tema da pesquisa. Este procedimento totalizou 41 artigos excluídos. Por fim, foram aceitos na revisão 11 artigos, 1 da SciElo, 6 da IEEE Xplore e 4 da MyEBSCO na segunda fase da revisão. Também foram adicionados ao estudo, em uma consulta prévia a revisão de literatura, 1 artigo, 3 livros, 1 dissertação, 1 tese e mais 2 documentos em páginas *web*, por serem trabalhos relevantes e estarem de acordo com o tema trabalhado na pesquisa, totalizando 19 textos na revisão de literatura. O Quadro 2 apresenta a lista dos textos da revisão.

Quadro 2. Lista de artigos selecionados na revisão

Lista de Artigos (Continua)			
Nº	**Título**	**Autores**	**Ano**
1	*Trajectory Graphs Appearing from the Skein Problems at the Hypercube*	Sagols *et al.*	2016
2	*Constellation Design with Hypercube Graphs*	Mishra *et al.*	2023
3	*Topology Design and Graph Embedding for Decentralized Federated Learning*	Duan *et al.*	2024
4	*Design and analysis of the rotational binary graph as an alternative to hypercube and Torus*	Seo; Lee.	2020
5	*Full Binary Tree-Based Fixed Degree Graph Design and Parallel Algorithm*	Seong *et al.*	2023
6	*The Characterizations of Hyper-Star Graphs Induced by Linearly Separable Boolean Functions*	Rao; Zhang.	2018
7	*A Characterization of Distance Matrices of Weighted Hypercube Graphs and Petersen Graphs*	Rubei; Ziani.	2020

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

8	*Set-to-Set Disjoint Paths Problem in Möbius Cubes*	Ichida; Kaneko.	2022

Lista de Artigos

(Conclusão)

Nº	**Título**	**Autores**	**Ano**
9	*1-Perfect Codes Over the Quad-Cube*	Jha.	2022
10	*Graph embedding strategy for efficient routing and optimal wavelength assignment in array-based WDM optical networks with enhanced hypercube topology*	Navis; Marappan.	2025
11	*The quantum hypercube as a k-mer graph*	Becerra-Gavino; Barbosa-Santillan.	2024

Fonte: Dos autores

O Quadro 3 apresenta a lista dos textos integrados de forma prévia a revisão de literatura.

Quadro 3. Lista de textos adicionais

Nº	**Título**	**Autores**	**Tipo**	**Ano**
1	Síntese de Algoritmos Paralelos para o n-cubo binário	Song.	Tese	1991
2	O Método do Desdobramento Recursivo para Imersão em Hipercubos e suas Aplicações	Takemoto.	Dissertação	1998
3	*Introduction to Graph Theory*	West.	Livro	1996
4	Grafos: Conceitos Fundamentais, Algoritmos e Aplicações	Paulo César Rodacki Gomes.	Livro	2022
5	*Hypercube*	Wolfram Research.	Página *web*	2025
6	*Hypercube*	Wolfram|Alpha.	Página *web*	2025
7	Introdução à Computação e Informação Quântica: Incluindo Álgebra Linear com Kets e Bras	Almeida.	Livro	2020
8	*Topological Indices, Graph Spectra, Entropies, Laplacians, and Matching Polynomials of n-Dimensional Hypercubes*	Balasubramanian.	Artigo	2023

Fonte: Dos autores

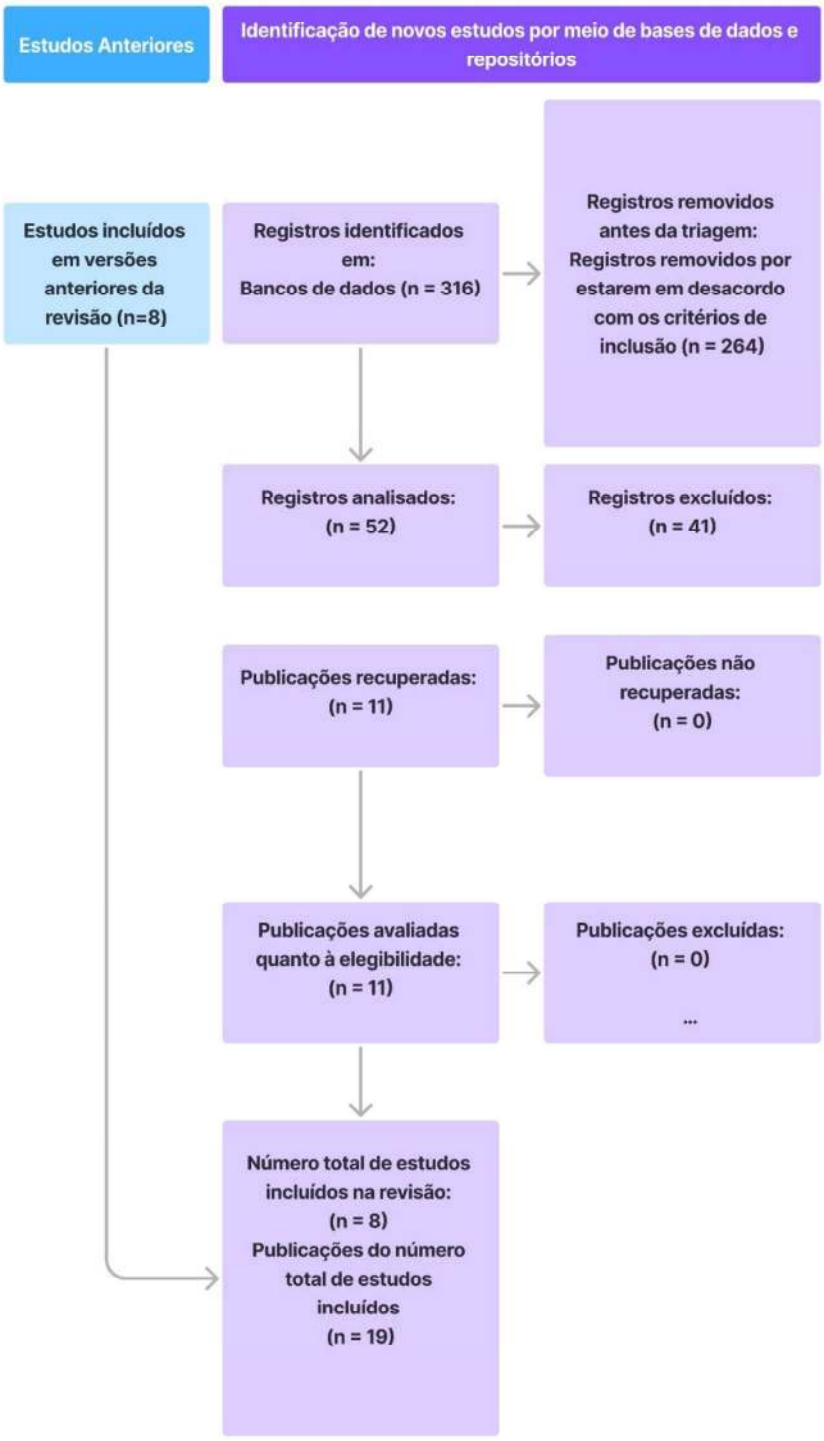
ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

A Figura 1 apresenta o diagrama do protocolo PRISMA para a pesquisa.

Figura 1. Diagrama do Protocolo PRISMA



Fonte: Dos autores

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

2.2. Análise

A seguir tem-se uma breve análise dos artigos de revisão integrativa deste trabalho de pesquisa, em que são investigados 11 artigos consultados em bases como a SciElo, IEEE Xplore e MyEBSCO. Além de um resumo dos assuntos tratados pelos artigos, é realizada uma menção de aplicações práticas de grafos hipercubos.

No trabalho *Trajectory Graphs Appearing from the Skein Problems at the Hypercube* dos autores Sagols, Feliú Morales-Luna, G. Buitrón-Dámaso de 2016. O artigo formaliza os *Skein Problems* em grafos hamiltonianos e demonstra sua redução ao Problema da Independência em Teoria dos Grafos. Destaca o uso potencial desses problemas em criptografia, especialmente em protocolos de autenticação de mensagens e identificação de entidades. O texto ainda apresenta uma boa conceitualização matemática com exemplos de aplicação computacional utilizando paradigma estruturado. No artigo os autores apresentam um algoritmo completo escrito em linguagem *open-source* Ruby para calcular trajetórias (*paths*) em grafos hipercubo. Este algoritmo apresenta baixa complexidade e pode ser transcrito para outras linguagens de programação como Python, por exemplo.

Constellation Design with Hypercube Graphs dos autores Kumar V. Mishra, Sunder Ram Krishnan, Brian M. Sadler, 2023. Este estudo aborda a redução de PAPR (*Peak-to-average power ratio*, relação potência pico-média) em sistemas de comunicação OFDM (*Orthogonal frequency division multiplexing*, multiplexação ortogonal por divisão de frequência), utilizando hipercubos para representar constelações de sinais. Propõe uma abordagem baseada em grafos para encontrar mapeamentos *Gray* ótimos, demonstrando redução média de 9–10 dB no PAPR com eficiência computacional. O artigo também apresenta boa fundamentação matemática dos grafos de hipercubos, além de exemplos gráficos e um algoritmo de heurística para encontrar um mapeamento de PAPR.

No artigo *Topology Design and Graph Embedding for Decentralized Federated Learning d*e Yubin Duan, Xiuqi Li e Jie Wu (2024), os autores abordam a otimização da eficiência de treinamento no Aprendizado Federado Descentralizado (DFL), no qual a sincronização dos modelos ocorre por comunicação dispositivo-a-dispositivo (D2D), eliminando o servidor central e reduzindo gargalos. O principal desafio é que a convergência depende fortemente da topologia da rede e da heterogeneidade dos dados. Para enfrentar esse problema, os autores propõem estratégias de construção de grafos baseadas em máxima similaridade, como o Hipercubo (HGC), utilizando o algoritmo de Blossom para pareamento perfeito, e Torus (TGC), por meio de um método heurístico, além de esquemas de sincronização como a Sincronização Sequencial (SS), que reduz o número de comunicações por rodada, e a Sincronização em Lote (*Batch*), que ajusta a frequência de comunicação via máscaras binárias.

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Os experimentos demonstram que a topologia de hipercubo apresenta convergência mais rápida devido ao maior intervalo espectral, superando anel e torus. Os métodos HGC e TGC reduziram a heterogeneidade efetiva entre vizinhos, alcançando desempenho próximo ao ótimo com complexidade polinomial. Como solução geral, os autores propõem modelar o problema como um *embedding* de grafos, combinando topologias de diâmetro logarítmico com esquemas de comunicação eficientes para tornar o DFL mais robusto em redes descentralizadas, como as baseadas em 5G. Além disso, o texto possui boa conceitualização matemática e apresenta algoritmos claros e passíveis de serem transcritos para linguagem Python.

Design and Analysis of the Rotational Binary Graph as an Alternative to Hypercube and Torus dos autores Junghyun Seo, HyeongOk Lee, 2020. Propõe o *Rotational Binary Graph* (RBG) como alternativa ao hipercubo e ao toro, combinando baixo diâmetro e grau constante. O estudo analisa propriedades topológicas, construção de árvores geradoras, custo de rede e algoritmo de *broadcast*, mostrando que o RBG oferece vantagens em escalabilidade e eficiência. O artigo faz ainda uma comparação de custos entre o design RBG e outras arquiteturas que se utilizam de objetos espaciais, além de exemplos de algoritmos e grafos. O texto apresenta três exemplos de algoritmos, o algoritmo 1 para construção de um BST (*Binary Spanning Tree*), algoritmo 2 para implementação do *Simple routing* e o algoritmo 3 para *Broadcasting*.

Em *Full Binary Tree-Based Fixed Degree Graph Design and Parallel Algorithm*, dos autores, Bo Ok Seong, Jin-Hyeok Jang e Hyeong Ok Lee (2023), os autores propõem o *Tree-baseGraph*, uma nova topologia de rede de interconexão para processamento paralelo baseada em uma árvore binária cheia com grau fixo de três. O diferencial desta estrutura é a inclusão de arestas adicionais conectando o nó raiz aos nós folha e as folhas entre si, o que reduz o diâmetro da rede para $2n-2$ e otimiza o custo de rede (produto entre grau e diâmetro). Comparado a topologias tradicionais como *mesh*, torus e *honeycomb mesh*, o *Tree-baseGraph* apresenta uma melhoria de cerca de 50% na eficiência de custo. Além do design estrutural, o estudo analisa algoritmos fundamentais para a operação da rede, incluindo um esquema de roteamento simples e um algoritmo recursivo para encontrar um ciclo Hamiltoniano. Os autores também comprovam a existência de três caminhos paralelos disjuntos por nós (denominados *Tree*, *Inside* e *Outside*), que aumentam a velocidade de transferência de dados e garantem tolerância a falhas, permitindo rotas alternativas caso um nó falhe. Os resultados validam o *Tree-baseGraph* como uma arquitetura robusta para computação de alto desempenho, superando as limitações de custo de *hardware* e processamento de modelos preexistentes. Ademais, o algoritmo proposto para a construção da topologia *Tree-baseGraph* apresenta implementação relativamente simples em Python, favorecendo sua adoção prática e reprodutibilidade experimental.

The Characterizations of Hyper-Star Graphs Induced by Linearly Separable Boolean Functions, dos autores Yanyi Rao; Xianda Zhang (2018), aborda a caracterização de grafos *hyper-star*, definidos como a união de vários hipercubos que compartilham pelo menos um vértice comum,

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

denominado fonte estelar (*star-source*). Os autores estabelecem uma relação fundamental ao demonstrar que toda função Booleana linearmente separável induz um subgrafo no hipercubo n-dimensional que possui obrigatoriamente a estrutura de um *hyper-star*. O estudo detalha propriedades métricas desses grafos, comprovando que eles são sempre conexos e que seu diâmetro é limitado pelo seu grau máximo, além de explorar a relação entre a fonte estelar e o grau dos vértices.

Para operacionalizar esse conceito, a pesquisa propõe três algoritmos principais: um para a decomposição de um *hyper-star* em seus sub-hipercubos máximos, um para o reconhecimento (determinação) de se um grafo é um *hyper-star* com complexidade de tempo e um algoritmo para a construção de todos os *hyper-stars* de n vértices. Os resultados indicam que essa abordagem baseada em teoria dos grafos é mais eficiente para o estudo e identificação de funções linearmente separáveis do que métodos computacionais anteriores, sendo útil para aplicações em redes neurais artificiais e teoria da confiabilidade.

A Characterization of Distance Matrices of Weighted Hypercube Graphs and Petersen Graphs de Elena Rubei, Dario Villanis Ziani, 2020. Caracteriza matrizes de distância de hipercubos ponderados e do grafo de Petersen, mostrando que nem todo grafo bipartido n-regular com vértices é necessariamente um hipercubo. O estudo contribui para análise métrica de grafos ponderados e relações topológicas específicas, além de trazer bons exemplos de aplicação computacional. O algoritmo apresentado no texto possui diversos comentários e é de fácil implementação, porém não está em linguagem Python, sendo necessário a conversão para esta linguagem de programação.

No artigo, *Set-to-Set Disjoint Paths Problem in Möbius Cubes*, dos autores Hiroyuki Ichida e Keiichi Kaneko (2022), tratam do problema de encontrar caminhos disjuntos por vértices entre dois conjuntos, S e T, em cubos de Möbius, uma topologia de rede de interconexão para sistemas paralelamente massivos. Os cubos de Möbius são destacados por apresentarem um diâmetro quase 50% menor do que o de um hipercubo comum com o mesmo número de nós, além de possuírem uma estrutura recursiva que facilita o desenvolvimento de algoritmos eficientes. A resolução do problema *set-to-set* é fundamental para garantir comunicações seguras, tolerância a falhas e aproveitamento total da largura de banda em supercomputadores.

A solução proposta é o Algoritmo S2S, que opera em três casos distintos dependendo da distribuição dos vértices de origem e destino nos sub-cubos da rede. O algoritmo apresenta uma complexidade de tempo de e garante que o comprimento máximo dos caminhos gerados seja de $2n-2$, superando o desempenho de algoritmos genéricos anteriores. Resultados de experimentos computacionais revelaram que, na prática, o algoritmo é ainda mais eficiente, confirmando sua utilidade prática em sistemas de alto desempenho.

Em *1-Perfect Codes Over the Quad-Cube*, do autor Pranava K. Jha (2022), o texto analisa a presença de códigos de correção de erro, conhecidos como 1-perfeitos, na topologia de rede

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

denominada *quad-cube*. O *quad-cube* é uma variação do hipercubo projetada para lidar com redes de altíssima escala, reduzindo a complexidade das conexões físicas sem perder a eficiência na comunicação e a tolerância a falhas. Esses códigos são fundamentais pois permitem que o sistema corrija automaticamente um erro simples em uma mensagem e detecte até dois erros, aumentando significativamente a confiabilidade das transmissões em supercomputadores e sistemas de processamento paralelo.

Além da segurança de dados, o estudo demonstra que esses códigos possibilitam o posicionamento ideal de recursos essenciais na rede, como fontes de energia e bibliotecas de dados. O autor comprova as condições necessárias para que a rede possa ser totalmente organizada através desses códigos, garantindo que cada usuário esteja sempre diretamente conectado a um único ponto de recurso. Essa organização otimiza o desempenho global do sistema e facilita a divisão da rede em agrupamentos menores e mais gerenciáveis, tornando o *quad-cube* uma arquitetura robusta para o mundo digital.

O artigo *Graph embedding strategy for efficient routing and optimal wavelength assignment in array-based WDM optical networks with enhanced hypercube topology*, de V. Vinitha Navis a, Raja Marappan (2025), aborda o desafio de otimizar o roteamento e a atribuição de comprimentos de onda em redes ópticas de alta velocidade, que utilizam a tecnologia de multiplexação para transmitir múltiplos canais de dados simultaneamente através de sinais de luz. Os autores propõem uma estratégia de incorporação de grafos que mapeia uma topologia de hipercubo aprimorado, superior ao hipercubo tradicional em termos de eficiência, custo e tolerância a falhas, em uma rede organizada de forma linear. Para isso, utilizam um esquema de numeração natural dos pontos de conexão para reduzir o congestionamento e um algoritmo que considera as limitações físicas do *hardware*, permitindo o reúso de canais de luz em caminhos que não se cruzam, o que minimiza a quantidade de recursos necessários para a transmissão.

Os resultados das simulações indicam que a solução proposta supera significativamente os métodos convencionais, garantindo uma alocação ideal de recursos com um tempo de processamento reduzido. O estudo revela uma descoberta importante: à medida que o tamanho da rede cresce, a necessidade de recursos da topologia aprimorada se torna quase idêntica à da versão comum, provando que é possível adotar estruturas mais robustas e rápidas sem custos adicionais de complexidade óptica em larga escala. Essa abordagem é altamente vantajosa para sistemas que exigem alta confiabilidade e baixa latência, como grandes centros de dados e supercomputadores, onde a eficiência no tráfego de dados e a resistência a falhas de componentes são críticas.

The quantum hypercube as a k-mer graph, de Gustavo Becerra-Gavino e Liliana Ibeth Barbosa-Santillan (2024), explora a interseção entre computação quântica e bioinformática, propondo o uso de um hipercubo quântico para representar e buscar fragmentos de DNA, conhecidos como *k-mers*. Nessa estrutura, as sequências genéticas são codificadas em formato

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

binário, permitindo que os estados dos *qubits* representem os vértices de um grafo onde a densidade de informação cresce de forma exponencial. Para realizar a busca de padrões marcados, os autores utilizam o algoritmo de caminhada quântica (*coined quantum walk search*), que aproveita propriedades quânticas para avaliar múltiplos caminhos simultaneamente no espaço de busca.

Os resultados demonstram que o sucesso da busca não é garantido em todos os casos, dependendo da combinação entre a inicialização do sistema e o alvo desejado. A análise classificou os resultados em diferentes categorias, revelando que apenas algumas configurações produzem padrões de acerto fortes e repetíveis, enquanto muitas outras resultam em saídas aleatórias. O estudo conclui que, embora a computação quântica prometa revolucionar a análise genômica pela sua alta capacidade de processamento paralelo, o algoritmo ainda carece de consistência universal, exigindo adaptações específicas para cada tipo de entrada para ser verdadeiramente prático na bioinformática.

2.3. Fundamentação teórica

Inicialmente, buscar-se-á apresentar os fundamentos teóricos da geometria do hipercubo, destacando sua construção a partir da generalização do quadrado e do cubo para dimensões superiores. Para fins de compreensão intuitiva, serão exploradas representações tridimensionais e projeções em dimensões inferiores, de modo a facilitar a visualização e interpretação de estruturas de ordem mais elevada. A seguir, será feita a caracterização dos grafos de hipercubo. Essa abordagem permitirá compreender o hipercubo não apenas como objeto geométrico, mas também como estrutura algébrica e combinatória, estabelecendo as bases necessárias para a investigação de suas propriedades e de suas possíveis aplicações em problemas de computação.

2.3.1. A Geometria do Hipercubo

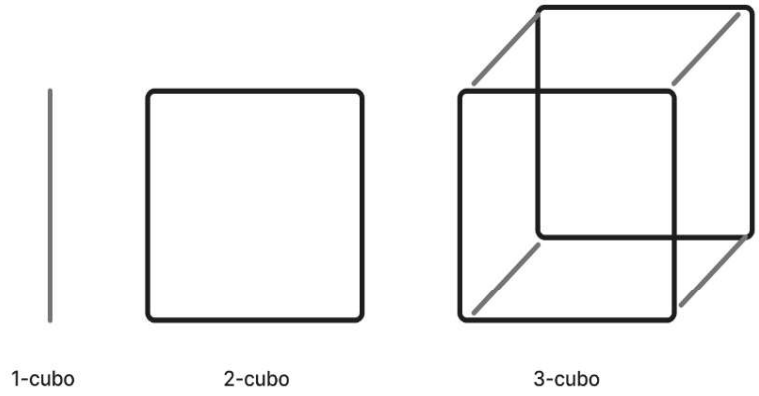
O hipercubo é uma generalização do 3-cubo para n dimensões, também chamado de $n-cubo$ ou polítopo de medida (WOLFRAM RESEARCH, 2025). Desta forma, para $n = 1$ tem-se um segmento de reta, para $n = 2$ tem-se um quadrado, para $n = 3$, um cubo, para $n = 4$, um tesserato, e assim por diante, como mostra a Figura 2.

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

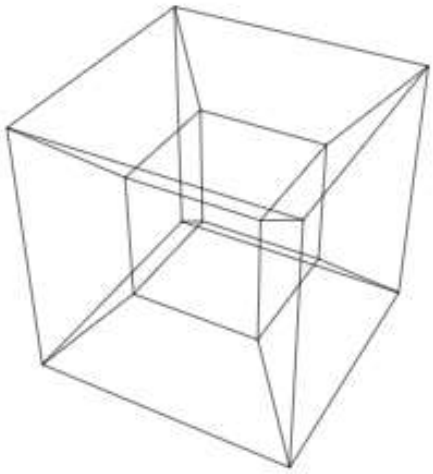
Figura 2. Hipercubos de 1, 2 e 3 dimensões



Fonte: Dos autores

Tem-se abaixo a Figura 3 do $4-cubo$ ou tesserato.

Figura 3. Hipercubo 4-cubo ou tesserato



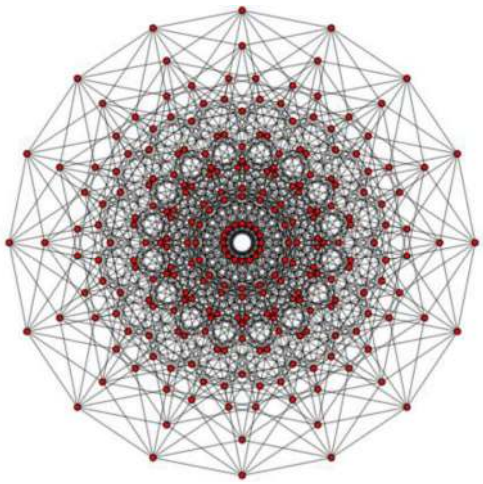
Fonte: WOLFRAM RESEARCH, (2025)

Na Figura 4 tem-se o grafo 8-cubo octaédrica do octâmero de água com 256 vértices e 1024 arestas.

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 4. Grafo 8-cubo do octâmero de água



Fonte: BALASUBRAMANIAN, (2023)

O $n-cubo$ possui V vértices, sendo que $V = 2^n$ indica a quantidade de vértices e $A = 2^{n-1}n$ a quantidade de arestas conectando seus vértices (WOLFRAM|ALPHA, 2025). Para calcular a quantidade de faces (quadrados) no hipercubo tem-se, $F = 2^{n-3}(n-1)n$.

Para calcularmos o número de cubos existentes em um hipercubo podemos utilizar a fórmula, $C = 2^{n-4}(n-1)(n-2)/3$. Aplicando-se, por exemplo, no tesserato as equações para cálculo de número de vértices, arestas, faces e cubos, tem-se, um total de 16 vértices, 32 arestas, 24 faces e 8 cubos.

2.3.2. Teoria dos Grafos

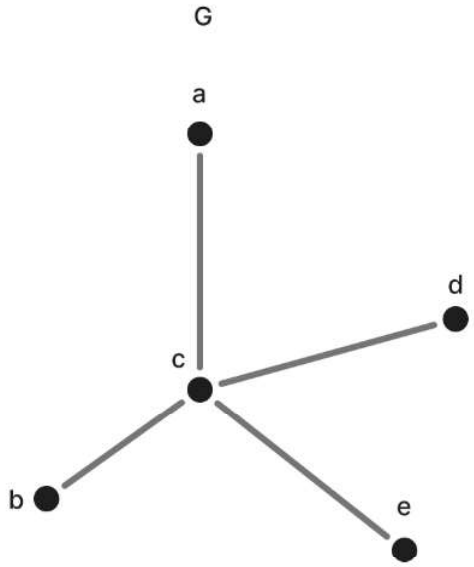
Um grafo pode ser entendido como uma abstração matemática que modela as relações entre elementos de determinados conjuntos, tomadas de dois em dois. Dessa forma, um grafo pode ser formalmente definido como um objeto $G = (V, E)$ para um conjunto não vazio V de vértices e um conjunto de arestas E em que cada aresta pertencente a E e pode ser compreendida como um par não-ordenado de vértices (u, v), com $(u, v) \in V$ (GOMES, 2022). Se $uv \in E(G)$ com $E(G) = \{e_1, \ldots e_m\}$ então u e v são adjacentes (WEST, 1996). Segundo o mesmo autor, um laço é uma aresta que conecta um vértice a ele mesmo, enquanto arestas paralelas ou múltiplas são aquelas que ligam o mesmo par de vértices. Um grafo simples é, portanto, aquele que não possui nem laços nem arestas múltiplas. Além disso, um grafo é denominado finito quando seus conjuntos de vértices e arestas contêm um número limitado de elementos (WEST, 1996). A Figura 5 mostra um exemplo de grafo simples.

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

Figura 5. Grafo G



Fonte: Dos autores

Em um grafo, considera-se que um par de vértices é conectável quando existe um caminho que une esses dois vértices (SAGOLS et al., 2016). Neste contexto, o hipercubo $Q_{(n)}$ pode ser definido como um grafo contendo um conjunto de vértices $\{V: V \subseteq [n]\}$ e um conjunto de arestas com todos os seus pares $\{V, V'\}$, em que $V \subseteq V'$ e $|V| = |V'| - 1$, com $[n] = \{1, \dots, n\}$ (AXENOVICH et al., 2024).

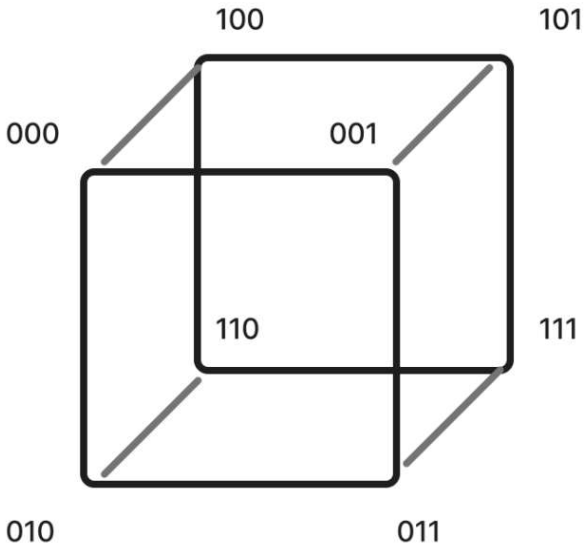
Sejam u e $v \in V$, $u = u_{n-1}u_{n-2} \dots u_1u_0$ e $v = v_{n-1}v_{n-2} \dots v_1v_0$ duas cadeias de $\bar{n}$ *bits*. Um hipercubo n-dimensional $Q_{(n)}$ de diâmetro n são todas as cadeias de $\bar{n}$ *bits* tidos como vértices. Além disso, u e v são considerados adjacentes, se e somente se, $h(u, v) = 1$ em que $h(u, v)$ é chamada de distância de Hamming entre u e v (SHYAMA; IYER, 2023).

O $n - cubo$ binário, ou ainda hipercubo de dimensão $\bar{n}$, possui N nós, de forma que $N = 2^n$ e cada nó pode ser identificado por um endereço de $\bar{n}$ *bits* (SONG, 1991). Para Takemoto (1998), uma rede implementada por hipercubo pode ser representada por um grafo em que os nós são rotulados por (*strings*) binárias com comprimento n, cujas arestas conectam dois nós com rótulos que diferem em uma posição de *bit*. Sendo assim, cada *bit* do rótulo representa uma dimensão do hipercubo. A Figura 6 busca clarificar a propriedade.

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 6. Hipercubo de dimensão 3 com endereçamento de *bits*



Fonte: Dos autores

A fundamentação na Teoria dos Grafos é essencial para responder à questão de pesquisa proposta, pois é nesse arcabouço matemático que o hipercubo se configura formalmente como um grafo cujos vértices representam cadeias binárias e cujas arestas conectam pares que diferem por exatamente um *bit*, segundo a distância de Hamming. Ao compreender conceitos como adjacência, conectividade, caminhos, diâmetro e estrutura de grafos simples, torna-se possível analisar o hipercubo como um caso particular de grafo regular e altamente simétrico, cujas propriedades topológicas favorecem aplicações computacionais. Assim, a investigação da geometria do hipercubo, à luz da Teoria dos Grafos, permite explorar suas características estruturais, como paralelismo, baixa distância entre nós e organização multidimensional, em simulações de grafos multidimensionais, contribuindo para modelagem, roteamento, processamento paralelo e outras aplicações em contextos computacionais.

3. METODOLOGIA

A abordagem metodológica adotada possui caráter teórico e dedutivo, fundamentando-se em uma revisão bibliográfica de autores das áreas de matemática discreta e da computação, combinada à apresentação e análise das propriedades do hipercubo e do grafo hipercubo. Complementarmente, desenvolveu-se uma análise de natureza teórico-computacional, que envolveu a transcrição de algoritmos para a linguagem Python e a realização de experimentos computacionais. Dessa forma, o procedimento metodológico buscou não apenas consolidar o

ISSN: 2675-6218 - RECIMA21
Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

embasamento teórico acerca do hipercubo e de sua representação por meio de grafos, mas também explorar suas potencialidades em contextos computacionais a partir da aplicação de algoritmos identificados na revisão de literatura. As simulações realizadas possibilitaram a análise das propriedades estruturais dos grafos hipercubos em diferentes contextos.

Para a modelagem e visualização desses objetos matemáticos, foi utilizada a linguagem Python 3, juntamente com as bibliotecas NetworkX, Pandas e Matplotlib, com suporte da IDE PyCharm. Os resultados obtidos nos experimentos computacionais foram organizados e disponibilizados em uma biblioteca de código em Python publicada na plataforma GitHub.

4. RESULTADOS E DISCUSSÃO

A seguir, são apresentados os resultados obtidos a partir dessas simulações computacionais.

4.1. Algoritmo *Trajectory Graphs*

O algoritmo *Trajectory Graphs* foi implementado em Python, usando o paradigma funcional, a partir de exemplos apresentados no artigo de Sagols *et al.* (2016), *Trajectory Graphs Appearing from the Skein Problems at the Hypercube*. Foram três arquivos de códigos, o trajectory_graphs.py, odd_hole.py e o arquivo main.py para gerenciar e executar os demais códigos. O resultado da execução do algoritmo pode ser visto na Figura 7.

Figura 7. Resultado da execução do código para *trajectory_graphs*



Fonte: Dos autores

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

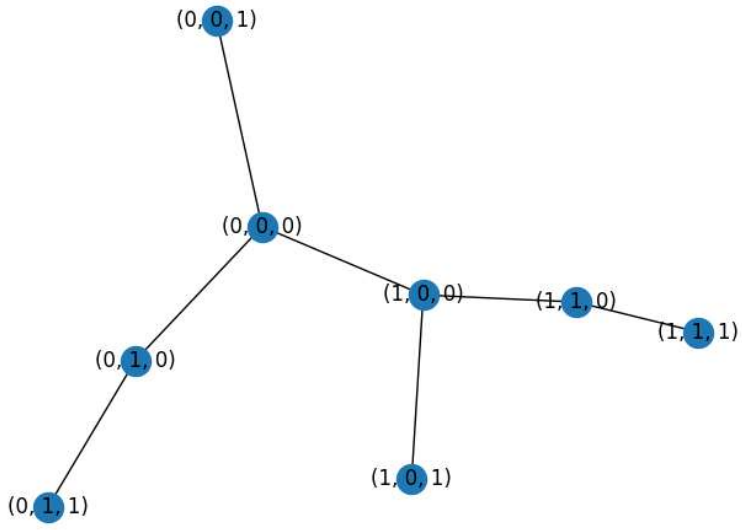
4.2. Algoritmo PAPR *Mapping*

O artigo de Kumar, Krishnan e Sadler (2023), ***Constellation Design with Hypercube Graphs***, também traz contribuições importantes como o exemplo do algoritmo para encontrar um mapeamento de baixo PAPR (relação potência pico-média). O algoritmo apresenta ainda uma arquitetura formada por dois arquivos, sendo a função de mapeamento implementada em papr.py e o arquivo main.py como o arquivo principal do programa.

4.3. Algoritmo *Rotational Binary Graphs*

O artigo *Design and Analysis of the Rotational Binary Graph as an Alternative to Hypercube and Torus* dos autores Junghyun Seo, HyeongOk Lee, (2020), também traz contribuições importantes para a computação com o algoritmo *Rotational Binary Graphs*. A arquitetura do programa é formada por quatro arquivos, binary_spaning_tree.py, broadcasting.py, simple_routing.py que implementam exatamente as soluções computacionais apresentadas no artigo e o arquivo main.py como o gerenciador da aplicação. Na Figura 8 tem-se o resultado da execução do programa de *Rotational Binary Graphs*.

Figura 8. Resultado da execução do programa *Rotational Binary Graphs*



Fonte: Dos autores

ISSN: 2675-6218 - RECIMA21
Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

4.4. Algoritmo *Distance Matrix*

O artigo *A Characterization of Distance Matrices of Weighted Hypercube Graphs and Petersen Graphs* de Elena Rubei, Dario Villanis Ziani (2020), trouxe o exemplo de cálculo para verificar se de D é a matriz de distâncias de um hipercubo onde todas as arestas são úteis.

A arquitetura da aplicação é formada por apenas 2 arquivos, o distance_matrix.py e o arquivo main.py.

4.5. Algoritmo *Max Similarity*

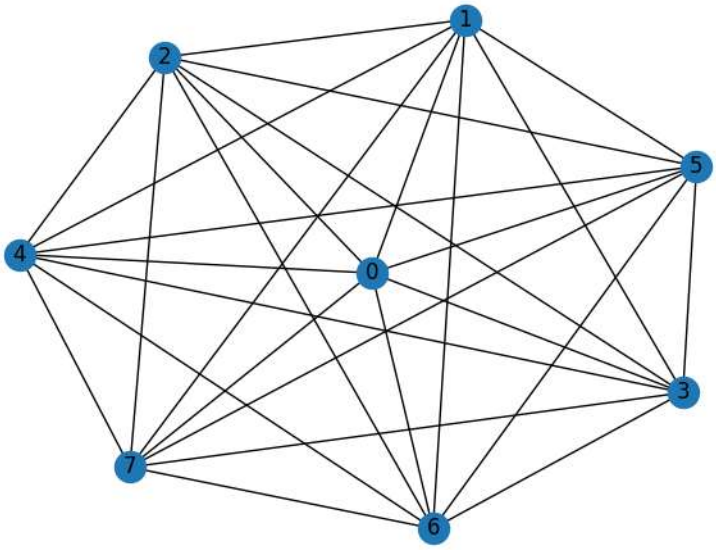
O artigo *Topology design and graph embedding for decentralized federated learning*, de Yubin Duan, Xiuqi Li e Jie Wu (2024), apresenta três algoritmos de *max-similarity*, para construção do hypercubo e do torus, além de um algoritmo do esquema da sequência de comunicação. O Algoritmo 1 (*Max-similarity hypercube construction*) constrói iterativamente um hipercubo d-dimensional a partir de um grafo completo com matriz de similaridade, agrupando os nós em *virtual nodes* e aplicando emparelhamento máximo ponderado (*Blossom*) para maximizar a similaridade total a cada dimensão adicionada. O Algoritmo 2 (*Max-similarity torus construction*) organiza os nós em uma estrutura de toro 2D, primeiro formando anéis locais com critério guloso de máxima similaridade e depois conectando esses anéis em um anel de anéis, resultando em uma topologia regular $m \times m$ que maximiza a similaridade global sob restrição planar periódica. Já o Algoritmo 3 (*Sequential communication scheme*) define um esquema determinístico de comunicação para um hipercubo d-dimensional, no qual, a cada iteração, cada nó sincroniza com seu vizinho correspondente a uma dimensão específica, determinada por $(i \bmod d)$. Esse mecanismo garante comunicação balanceada, cobertura completa das dimensões após d iterações e eficiência logarítmica no diâmetro da rede, sendo especialmente adequado para treinamento distribuído, sincronização paralela e operações do tipo *AllReduce* em arquiteturas de alto desempenho. A Figura 9 mostra a construção do hipercubo de máxima similaridade.

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

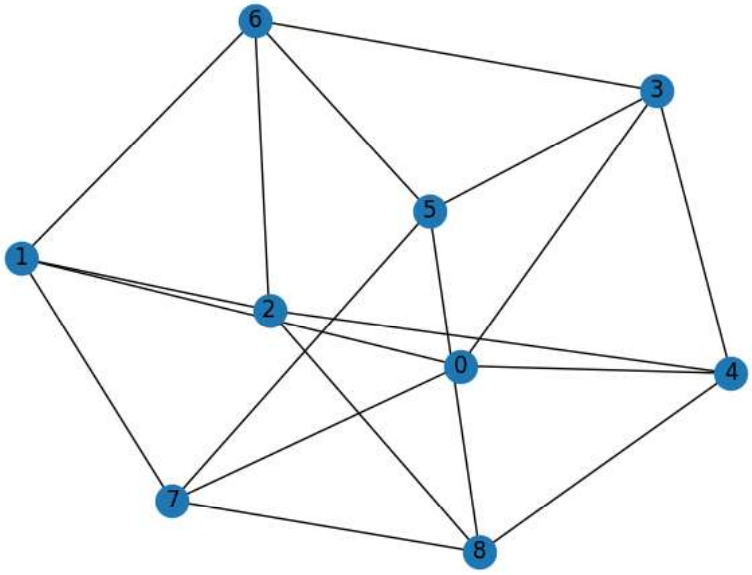
Figura 9. Construção do hipercubo de máxima similaridade



Fonte: Dos autores

A Figura 10 apresenta a construção gráfica do torus de máxima similaridade.

Figura 10. Construção do torus de máxima similaridade



Fonte: Dos autores

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

4.6. *Parallel Algorithm*

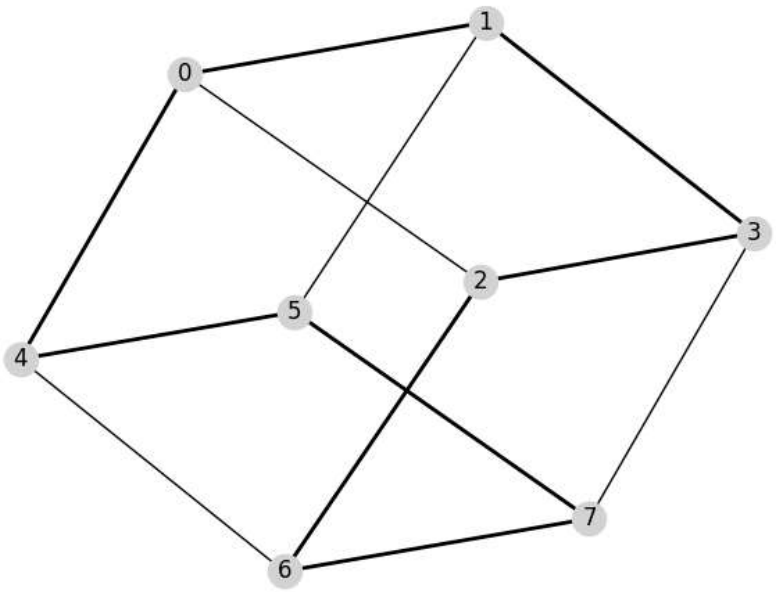
O artigo *Full Binary Tree-Based Fixed Degree Graph Design and Parallel Algorithm*, de Bo Ok Seong, Jin-Hyeok Jang e Hyeong Ok Lee (2023), traz cinco algoritmos, *simple_routing*, *Hamiltonian cycle algorithm*, *parallel_path_algorithm*, *Inside(A, a)* e *Outside(A, a)*. Estes algoritmos são estratégias de roteamento em estruturas hierárquicas como árvores binárias implícitas ou redes derivadas de hipercubos. O *Simple Routing* realiza o caminho clássico entre dois nós subindo até o ancestral comum mais próximo e depois descendo até o destino, garantindo complexidade $O(\log N)$. O *Parallel Path* otimiza essa ideia avaliando previamente, por meio de uma verificação estrutural (como *Check*), qual direção produz menor expansão antes de aplicar o roteamento, podendo inverter a ordem do percurso para melhorar eficiência. Já os algoritmos *Inside* e *Outside* exploram estratégias complementares: o *Inside* mantém o roteamento dentro da subárvore que contém o destino, expandindo até um nível máximo e ajustando internamente antes de contrair; o *Outside*, por sua vez, sai deliberadamente da subárvore atual, sobe até níveis superiores (ou até a raiz/borda estrutural), percorre externamente e depois reentra na região do destino.

O *Hamiltonian Cycle* difere dos demais por não ser apenas um algoritmo de roteamento ponto-a-ponto, mas um método para construir um ciclo que visita todos os nós exatamente uma vez antes de retornar ao início, sendo amplamente utilizado em grafos como hipercubos e redes regulares para balanceamento e cobertura completa da topologia. Enquanto *Simple Routing* e *Parallel Path* focam em caminhos eficientes entre dois nós específicos, e *Inside/Outside* exploram decisões estruturais baseadas em intervalos binários e níveis hierárquicos, o *Hamiltonian Cycle* garante uma travessia global ótima em termos de cobertura, frequentemente usada como base para estratégias de comunicação paralela e escalonamento em arquiteturas distribuídas. Na Figura 11 é o resultado da execução do algoritmo *hamiltonian_cycle*.

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 11. Resultado da execução de *hamiltonian_cycle*



Fonte: Dos autores

4.7. Algoritmo *Hyper-Star*

The Characterizations of Hyper-Star Graphs Induced by Linearly Separable Boolean Functions, de RAO Yanyi e ZHANG Xianda (2018), apresenta três algoritmos para construção da *hiper-star*. Os algoritmos tratam da identificação, decomposição e expansão estrutural desse tipo específico de grafo, que é definido por vértices binários com peso fixo (mesmo número de *bits* 1) e arestas determinadas por trocas envolvendo uma posição central. O *DecompositionHyperStar* particiona uma *Hyper-Star* em componentes menores, escolhendo iterativamente vértices de grau mínimo, formando subconjuntos conectados com o mesmo grau mínimo e removendo-os do grafo até que ele fique vazio. Esse processo revela a estrutura interna do grafo e permite identificar subcomponentes homogêneos.

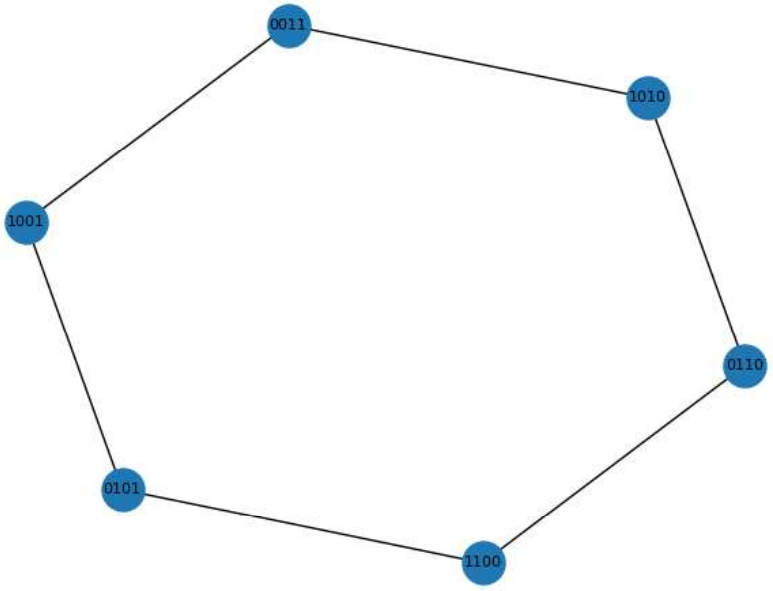
O algoritmo *IsHyperStar* verifica se um grafo dado possui a estrutura característica de uma *Hyper-Star*. Ele procura um vértice que satisfaça a propriedade estrutural central: para todos os outros vértices, deve existir a relação estrutural específica que define a adjacência na *Hyper-Star* (como a troca envolvendo uma posição fixa). Caso tal vértice exista, o grafo é reconhecido como uma *Hyper-Star*. Esse procedimento funciona como um teste de reconhecimento estrutural.

Já o *ExtendHyperStars* constrói novas *Hyper-Stars* a partir de subconjuntos existentes, adicionando vértices candidatos e verificando se o subgrafo resultante permanece conectado e mantém a propriedade de *Hyper-Star*. Após gerar possíveis extensões, o algoritmo remove estruturas isomorfas para evitar duplicações. Em conjunto, esses três algoritmos permitem

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

decompor, reconhecer e expandir *Hyper-Stars*, sendo úteis na análise estrutural e enumeração de subgrafos em redes combinatórias. A Figura 12 mostra o grafo resultante do algoritmo *decomposition hyper-star*.

Figura 12. Algoritmo *Decomposition Hyper-Star*



Fonte: Dos autores

4.8. Algoritmo *Mobius Cube*

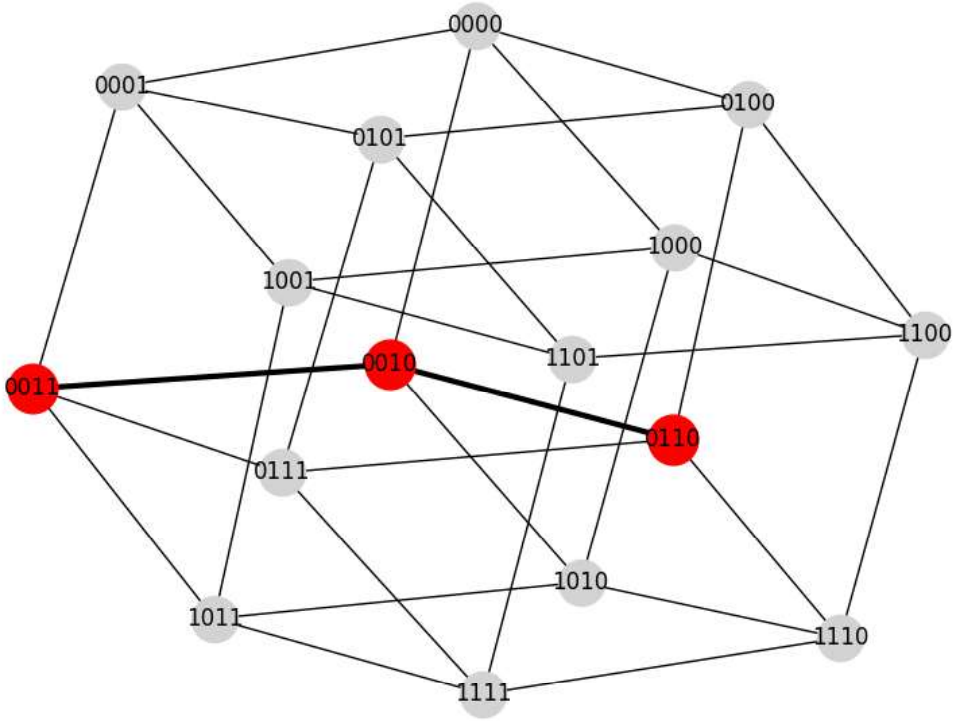
Set-to-Set Disjoint Paths Problem in Möbius Cubes, de Hiroyuki Ichida e Keiichi Kaneko (2022), apresenta o algoritmo SP (*Shortest Path*) para hipercubos que calcula um caminho mínimo entre dois vértices s e t explorando a representação binária dos nós. Ele utiliza a operação XOR ($w = s \oplus t$) para identificar quais *bits* diferem entre os dois vértices, pois cada *bit* diferente indica uma dimensão em que é necessário realizar uma transição. O algoritmo percorre essas posições de *bits* e decide quais devem ser invertidas (*flip*) seguindo uma regra sequencial, construindo um conjunto de dimensões a serem alteradas. Em seguida, aplica essas inversões ordenadamente a partir do vértice inicial, garantindo que cada passo mude exatamente um *bit*, o que corresponde a percorrer uma aresta do hipercubo. Como cada mudança reduz a diferença binária entre s e t, o caminho gerado tem comprimento igual à distância de Hamming entre os vértices, assegurando que seja um caminho mínimo. A Figura 13 mostra o resultado da execução do algoritmo.

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

Figura 13. Algoritmo *Mobius Cube*



Fonte: Dos autores

4.9. Algoritmo *k-mer graph*

The quantum hypercube as a k-mer graph, de Gustavo Becerra-Gavino e Liliana Ibeth Barbosa-Santillan (2024), apresenta um algoritmo para representar sequências de DNA (*k-mers*) como vértices de um hipercubo quântico, explorando a superposição para busca paralela de padrões. Os códigos apresentados implementam essa ideia em três etapas principais dentro da biblioteca *Qiskit*. O algoritmo 1 define o *shift operator* do hipercubo 4D, responsável por realizar as transições entre vértices (estados) que diferem por um *bit*, usando portas X e Toffoli (CCX) para simular a movimentação da caminhada quântica entre nós adjacentes. O algoritmo 2 inicializa o circuito quântico com uma *string* binária específica, codificando o estado inicial que representa determinado *k-mer* no espaço do hipercubo.

Já o algoritmo 3 implementa o operador de marcação (*oracle*), que identifica o padrão alvo dentro da superposição. Ele aplica portas X condicionais para ajustar os *qubits* correspondentes aos *bits* 0 do alvo, seguida de uma porta multi-controlada (MCT) para inverter a fase do estado marcado, característica fundamental do algoritmo de busca por caminhada quântica (*coined quantum walk*

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

search). Em conjunto, esses blocos constroem o mecanismo de busca no hipercubo quântico: inicialização do estado, evolução (caminhada) e marcação do padrão desejado, permitindo explorar paralelamente múltiplos caminhos no espaço combinatório dos *k-mers*. Embora o código original já houvesse sido implementado em linguagem Python, os autores o disponibilizaram no repositório do GitHub: https://github.com/dti-data/quantum-k-mer-graph/tree/main, assim não foi necessário o desenvolvimento de um novo programa com a transcrição do algoritmo.

4.10. Algoritmo *Routing and Wavelength Assignment*

No artigo *Graph embedding strategy for efficient routing and optimal wavelength assignment in array-based WDM optical networks with enhanced hypercube topology*, de V. Vinitha Navis e Raja Marappan (2025), os autores apresentam um algoritmo para o modelo estático RWA (*Routing and Wavelength Assignment*). O algoritmo realiza o cálculo e a atribuição de comprimentos de onda (λ) às arestas do hipercubo, considerando restrições estruturais e de *hardware*. Inicialmente, ele calcula o conjunto de valores possíveis de λ com base na paridade de n e r, por meio das funções auxiliares w, calculate_w1 e calculate_w2, que geram potências de 2 conforme regras combinatórias. Em seguida, o algoritmo constrói ciclos específicos no hipercubo e tenta atribuir a cada ciclo um λ disponível, verificando três condições principais: se os ciclos são disjuntos, se o λ não ultrapassa o limite máximo por fibra e se o número de portas por vértice não excede o permitido. Caso nenhuma fibra existente suporte a nova atribuição, uma nova fibra é criada. O processo garante uma distribuição válida de λ respeitando simultaneamente as restrições topológicas do hipercubo e as limitações físicas da infraestrutura óptica.

4.11. Algoritmo *Quad-Cube*

Por fim, o artigo *1-Perfect Codes Over the Quad-Cube*, do autor Pranava K. Jha (2022), nos trouxe sete algoritmos para construção do *Quad-cube*. Os Algoritmos 1 a 7 descrevem uma construção hierárquica e recursiva baseada em hipercubos e partições estruturadas do espaço binário, utilizando deslocamentos por potências de 2 para organizar subconjuntos de forma perfeitamente alinhada. O algoritmo 1 constrói partições em códigos de Hamming dentro do hipercubo e que servem como blocos fundamentais. A partir dele, os algoritmos 2 a 6 expandem essa estrutura por meio de laços aninhados que introduzem novos níveis de organização formando partições cada vez maiores através de somas estruturadas. Cada novo algoritmo corresponde a uma ampliação da profundidade combinatória, preservando regularidade e garantindo que os conjuntos resultantes sejam disjuntos e cubram completamente o espaço considerado.

O algoritmo 7 generaliza todo o esquema ao introduzir um parâmetro adicional t, permitindo gerar famílias completas de construções mantendo a cardinalidade estrutural. Em todos os casos, o crescimento do tamanho dos conjuntos é exponencial, refletindo a natureza combinatória do

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

hipercubo e dos códigos envolvidos. A sequência de algoritmos mostra como uma partição inicial baseada em códigos de Hamming pode ser sistematicamente elevada para estruturas de grandes dimensão, mantendo coerência algébrica e simetria binária ao longo de todo o processo.

5. ANÁLISE E DISCUSSÃO DOS RESULTADOS

Diversos algoritmos relacionados a grafos hipercubo, descritos na literatura científica, foram transcritos e simulados em linguagem Python com o objetivo de avaliar suas propriedades computacionais e potencial de aplicação. Entre eles, destacam-se os algoritmos *trajectory_graphs* e *odd_hole*, baseados no trabalho de Sagols *et al.* (2016), que permitem gerar hipercubos de n dimensões, construir trajetórias no grafo e identificar ciclos ímpares. Os resultados foram apresentados por meio de visualizações gráficas, evidenciando a viabilidade da implementação computacional dessas estruturas.

Também foram implementados algoritmos voltados ao mapeamento e à organização estrutural de grafos hipercubo, como o método de *low PAPR mapping*, que utiliza uma matriz de permutação aproximada para realizar o mapeamento de *bits*, e o *Rotational Binary Graph*, responsável pela geração de uma árvore geradora binária em um hipercubo. Este último incluiu ainda rotinas adicionais de roteamento e *broadcasting* entre nós, sendo um dos algoritmos mais complexos implementados e apresentando bons resultados em termos de execução e visualização gráfica.

Outros estudos analisados incluem algoritmos de roteamento paralelo, geração de *hyper-star graphs* e busca de caminhos mínimos em *Möbius cubes*, todos transcritos para Python e, em alguns casos, enriquecidos com recursos de visualização gráfica para facilitar a análise estrutural dos grafos. Além disso, foram explorados trabalhos relacionados à computação quântica, como a representação de sequências de DNA em hipercubos quânticos utilizando a biblioteca Qiskit da IBM, bem como algoritmos voltados à otimização de redes ópticas e ao cálculo de matrizes de distância em hipercubos. Neste caso, o algoritmo necessita de aprimoramentos para alcançar melhores resultados em computação quântica.

Também foram implementados algoritmos associados à construção de códigos perfeitos em *quad-cubes*, que apresentam potencial aplicação em segurança da informação e codificação de dados. De modo geral, os experimentos indicam que as estruturas baseadas em hipercubos possuem amplo potencial de aplicação em áreas como inteligência artificial, comunicação em redes, computação quântica e representação do conhecimento, especialmente quando integradas a modelos formais como ontologias, taxonomias e *Linked Data*.

Com exceção do algoritmo *k-mer graph*, que não foi transcrito para Python, e dos algoritmos *PAPR Mapping*, *Distance Matrix*, *Routing and Wavelength Assignment* e *Quad-Cube*, que não apresentaram resultados gráficos satisfatórios, todos os demais algoritmos foram adaptados para

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

incluir rotinas de renderização dos resultados utilizando a biblioteca Matplotlib. Essa adaptação permitiu visualizar graficamente as estruturas e os caminhos gerados nos grafos, contribuindo para a análise das propriedades dos modelos implementados.

Como perspectiva para trabalhos futuros, recomenda-se o uso de bibliotecas gráficas mais robustas, como PyVista, que oferecem recursos avançados de visualização tridimensional e melhor desempenho na renderização de estruturas complexas e de maior dimensionalidade. Essas ferramentas podem ampliar significativamente a qualidade da visualização e facilitar a análise de grafos hipercubo em contextos científicos e computacionais mais exigentes.

Por fim, recomenda-se o desenvolvimento de um serviço que disponibilize soluções práticas e de código aberto à comunidade científica. O código completo dos algoritmos encontra-se disponível no repositório GitHub: https://github.com/ivamgf/hypercube-algoritms, sob licença Creative Commons Zero v1.0 Universal e podem ser acessadas sem restrições.

6. CONSIDERAÇÕES

Este trabalho buscou investigar de que forma a geometria do hipercubo e as propriedades dos grafos de hipercubo podem ser exploradas em aplicações computacionais. Para isto foram transcritos para linguagem Python diversos algoritmos encontrados na literatura.

A partir da revisão de literatura, foram analisados onze textos selecionados, um disponível na SciELO, seis na IEEE Xplore e quatro na MyEBSCO, além de outros trabalhos complementares. A revisão permitiu identificar as principais propriedades matemáticas do hipercubo, sua representação como grafo e as aplicações existentes em computação, redes de interconexão e análise de dados complexos.

Complementarmente, os resultados obtidos por meio das simulações computacionais apresentaram hipercubos de diferentes dimensões, com grafos devidamente renderizados utilizando a biblioteca matplotlib. Neste trabalho foram desenvolvidas seis soluções computacionais sugeridas pela literatura utilizando linguagem Python como os algoritmos de *Rotational Binary Graphs, Max Similarity, Parallel Algorithm, Hyper-Star, Mobius Cube* e o algoritmo de *Trajectory Graphs*. Esses experimentos reforçam o potencial do hipercubo como ferramenta matemática e computacional para a modelagem e a resolução de problemas complexos, tais como criptografia, sistemas de comunicação e técnicas de *embeddings*. Ademais, a biblioteca desenvolvida neste estudo configura-se como uma contribuição relevante, ao disponibilizar um conjunto de algoritmos funcionais e reutilizáveis que podem apoiar outros pesquisadores na exploração da arquitetura de hipercubos em diferentes contextos científicos e tecnológicos.

Como recomendação para pesquisas futuras, sugere-se o aprofundamento das simulações computacionais envolvendo hipercubos, bem como a implementação de algoritmos que viabilizem a aplicação prática das soluções identificadas na revisão de literatura.

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Adicionalmente, propõe-se uma mudança de paradigma na construção do algoritmo, adotando-se a orientação a objetos. Essa abordagem permitiria que as propriedades e os métodos definidos para o hipercubo fossem reutilizados e herdados por outros algoritmos, favorecendo maior modularidade, reutilização de código e escalabilidade. Esse avanço possibilitaria a consolidação de um modelo mais robusto para a exploração de grafos hipercubo em aplicações práticas, ampliando sua aplicabilidade e promovendo uma integração mais consistente entre a teoria matemática e a simulação computacional.

Agradecimentos

Um agradecimento especial à Universidade Federal de Santa Catarina e ao Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento e ao laboratório Lamid pelo apoio dado a pesquisa que resultou neste artigo e à CAPES pelo financiamento da bolsa de pesquisa que permitiu a realização deste trabalho.

Declaração de uso de Inteligência Artificial

Os autores declaram que ferramentas de Inteligência Artificial generativa foram utilizadas de forma limitada, como apoio à revisão linguística do manuscrito, incluindo correções gramaticais, ortográficas e de estilo, bem como no aprimoramento e na revisão do resumo. O uso dessas ferramentas teve finalidade exclusivamente editorial, visando melhorar a clareza e a qualidade da redação do texto. As ferramentas de IA não foram utilizadas para a geração de resultados científicos, a análise de dados, a elaboração de métodos ou a formulação de conclusões. Todas as decisões relacionadas ao conteúdo científico, interpretação dos resultados e estrutura final do manuscrito foram realizadas pelos autores, que assumem integral responsabilidade pela precisão, originalidade e integridade do trabalho apresentado.

REFERÊNCIAS

ALMEIDA, Norton Gomes de. **Introdução à Computação e Informação Quântica:** Incluindo Álgebra Linear com Kets e Bras. 1° ed. São Paulo: Livraria da Física, 2020.

BALASUBRAMANIAN, Krishnan. Topological indices, graph spectra, entropies, Laplacians, and matching polynomials of n-dimensional hypercubes. **Symmetry**, v. 15, n. 2, p. 557, 2023.

BECERRA-GAVINO, Gustavo; BARBOSA-SANTILLAN, Liliana Ibeth. The quantum hypercube as a k-mer graph. **Frontiers in Bioinformatics**, v. 4, p. 1401223, 2024.

DUAN, Yubin; LI, Xiuqi; WU, Jie. Topology design and graph embedding for decentralized federated learning. **Intelligent and Converged Networks**, v. 5, n. 2, p. 100-115, 2024.

GOMES, Paulo César Rodacki. **Grafos:** Conceitos Fundamentais, Algoritmos e Aplicações. Blumenau: IFC, 2022.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

ICHIDA, Hiroyuki; KANEKO, Keiichi. Set-to-set disjoint paths problem in Möbius cubes. **IEEE Access**, v. 10, p. 83075-83084, 2022.

JHA, Pranava K. 1-perfect codes over the quad-cube. **IEEE Transactions on Information Theory**, v. 68, n. 10, p. 6481-6504, 2022.

MISHRA, Kumar Vijay; KRISHNAN, Sunder Ram; SADLER, Brian M. *Constellation design with hypercube graphs*. **IEEE Signal Processing Letters**, v. 30, p. 1152–1156, 2023. Disponível em: https://doi.org/10.1109/LSP.2023.3308813. Acesso em: 17 ago. 2025.

NAVIS, V. Vinitha; MARAPPAN, Raja. Graph embedding strategy for efficient routing and optimal wavelength assignment in array-based WDM optical networks with enhanced hypercube topology. **Computers and Electrical Engineering**, v. 127, p. 110571, 2025.

RAO, Yanyi; ZHANG, Xianda. *The characterizations of hyper-star graphs induced by linearly separable Boolean functions*. **Chinese Journal of Electronics**, v. 27, n. 1, p. 19–25, 2018.

RUBEI, Elena; ZIANI, Dario Villanis. A characterization of distance matrices of weighted hypercube graphs and Petersen graphs. **Journal of Multiple-Valued Logic & Soft Computing**, v. 34, 2020. Disponível em: <https://research.ebsco.com/c/spunt7/viewer/pdf/vkkwjy37rz>. Acesso em: 3 abr. 2026.

SAGOLS, Feliú; MORALES-LUNA, Guillermo; BUITRON-DAMASO, Israel. Trajectory Graphs Appearing from the Skein Problems at the Hypercube. **Computación y Sistemas**, v. 20, n. 1, p. 81-87, 2016. https://doi.org/10.13053/cys-20-1-2061. Acesso em: 17 ago. 2025.

SEO, Jung-hyun; LEE, HyeongOk. Design and analysis of the rotational binary graph as an alternative to hypercube and Torus. **The Journal of Supercomputing**, v. 76, n. 9, p. 7161-7176, 2020. https://doi.org/10.1007/s11227-019-03115-x. Acesso em: 17 ago. 2025.

SEONG, Bo Ok; JANG, Jin-Hyeok; LEE, Hyeong Ok. Full Binary Tree-Based Fixed Degree Graph Design and Parallel Algorithm. **IEEE Access**, v. 11, p. 120816-120826, 2023.

SONG, Siang Wun. **Síntese de algoritmos paralelos para o n-cubo binário**. 1991. Tese de Doutorado. Universidade de São Paulo.

TAKEMOTO, Carla Yayoi. **O método de Dobramento Recursivo para Imersão em Hipercubos e suas Aplicações**. 1998. Tese de Doutorado. Universidade de São Paulo.

WEST, Douglas Brent et al. **Introduction to graph theory**. Upper Saddle River: Prentice hall, 1996.

WOLFRAM RESEARCH. **Hypercube**. Disponível em: https://mathworld.wolfram.com/Hypercube.html. Acesso em: 5 out. 2025.

WOLFRAM|ALPHA. **Hypercube**. Disponível em: https://www.wolframalpha.com/input/?i=hypercube. Acesso em: 5 out. 2025.

ISSN: 2675-6218 - RECIMA21

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.