



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE TELECOMUNICAÇÕES

MATHEUS DE ALMEIDA ESTEBAN

BENCHMARKING DE ALGORITMOS CLÁSSICOS E VARIACIONAIS QUÂNTICOS
PARA O PROBLEMA MAXCUT EM REDES DE TELECOMUNICAÇÕES

FORTALEZA

2026

MATHEUS DE ALMEIDA ESTEBAN

BENCHMARKING DE ALGORITMOS CLÁSSICOS E VARIACIONAIS QUÂNTICOS
PARA O PROBLEMA MAXCUT EM REDES DE TELECOMUNICAÇÕES

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Telecomunicações do Centro de Tecnologia da Universidade Federal do Ceará, como requisito para obtenção do grau de bacharel em Engenharia de Telecomunicações.

Orientador: Prof. Dr. João Batista Rosa Silva

FORTALEZA

2026

MATHEUS DE ALMEIDA ESTEBAN

BENCHMARKING DE ALGORITMOS CLÁSSICOS E VARIACIONAIS QUÂNTICOS
PARA O PROBLEMA MAXCUT EM REDES DE TELECOMUNICAÇÕES

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Telecomunicações do Centro de Tecnologia da Universidade Federal do Ceará, como requisito para obtenção do grau de bacharel em Engenharia de Telecomunicações.

Aprovada em: 06/07/2026

BANCA EXAMINADORA

Prof. Dr. João Batista Rosa Silva (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Joacir Soares de Andrade
Universidade Federal do Ceará (UFC)

Prof. Dr. Kleber Zuza Nóbrega
Universidade Federal do Ceará (UFC)

À minha família e aos meus amigos, por sempre
acreditarem em mim e por todo o apoio durante
esta jornada.

AGRADECIMENTOS

Agradeço, primeiramente, aos meus pais, André e Liliane, que sempre foram meu esteio, oferecendo todo o apoio e incentivo necessários para que eu jamais desistisse dos meus objetivos e me tornasse quem sou hoje. Sou grato a toda a minha família, cujo suporte foi fundamental para esta conclusão.

Gostaria de prestar uma homenagem ao Dr. Ionilton Pereira do Vale (in memoriam) e à sua filha, Ana Lídia Alves, que me acolheram como se eu fosse parte da família. Mesmo com o passar do tempo e com as mudanças, o apoio de vocês permanece como uma marca profunda em minha trajetória pessoal.

Aos meus amigos e colegas de faculdade, em especial aos que compartilharam comigo a vivência no DAAM e na Célula de IoT — Douglas, Lais, Wanderson, Israel, Mateus, Augustinho, Gabriel, Bosco, e Ivo —, agradeço pelas conversas diárias, pela parceria constante e por tornarem a rotina acadêmica um ambiente de aprendizado mais divertido. Agradeço também aos meus amigos Erick, Celina, Denilson, Wesley e João Gabriel pela amizade e por estarem presentes ao longo de toda esta jornada.

Aos professores da Universidade Federal do Ceará, agradeço pela dedicação em compartilhar conhecimento e por contribuírem de forma significativa para minha formação acadêmica e profissional.

Ao meu orientador, prof. João Batista, e ao meu amigo, Orleans Cardoso, agradeço pela confiança e dedicação ao longo de todo o processo de orientação. Agradeço, ainda, pelos incentivos constantes, que foram essenciais para a condução desta pesquisa, especialmente nos momentos mais desafiadores.

Por fim, reconheço a importância de todos que, direta ou indiretamente, contribuíram com ideias, técnicas e ajuda neste trabalho, incluindo aqueles que desenvolveram e adaptaram os modelos de documentação necessários para a padronização acadêmica. A cada um de vocês que, de alguma forma, participaram esta história, o meu sincero obrigado.

“O acaso só favorece a mente preparada.”

(Louis Pasteur)

RESUMO

O problema do corte máximo (*Maximum Cut* - MaxCut) é um problema de tempo Polinomial Não Determinístico (*Nondeterministic Polynomial time* - NP)-difícil com aplicações relevantes em cenários de particionamento e alocação de recursos em redes de telecomunicações. Neste trabalho, é apresentado um benchmarking entre três abordagens para sua resolução: a solução exata por força bruta, a aproximação clássica baseada no algoritmo de Goemans-Williamson (*Goemans-Williamson algorithm* - GW), abordagem variacional quântica por meio do Solucionador Quântico Variacional (*Variational Quantum Eigensolver* - VQE) e o Algoritmo de Otimização Quântica Aproximada (*Quantum Approximate Optimization Algorithm* - QAOA). Para validar a viabilidade destas metodologias em cenários práticos, o trabalho incorpora um estudo de caso focado na otimização estrutural de uma rede de telecomunicações onde os algoritmos foram aplicados. Os resultados demonstram que o algoritmo clássico de Goemans-Williamson oferece o melhor custo-benefício, mantendo alta eficiência temporal e AR próxima a 1,0000. Entre os modelos quânticos, evidenciou-se um *trade-off*: o VQE apresentou maior qualidade da solução à custa de um espaço paramétrico de ordem $\mathcal{O}(N)$, enquanto o QAOA alcançou excelente compactação paramétrica na ordem de $\mathcal{O}(p)$, porém com degradação na qualidade do corte à medida que o problema escalou. Conclui-se que a principal barreira atual para a vantagem quântica nessas formulações não reside na matemática dos *ansätze*, mas no custo exponencial da simulação clássica de vetores de estado, reforçando a necessidade de execução em hardwares quânticos reais para a obtenção de ganhos práticos.

Palavras-chave: MaxCut. Grafos. VQE. Algoritmo de Goemans-Williamson. Computação Quântica. Qiskit

ABSTRACT

The maximum cut (MaxCut) problem is a Non-Deterministic Polynomial-time (NP)-hard problem with relevant applications in partitioning and resource allocation scenarios in telecommunications networks. In this work, a benchmarking among three approaches for its resolution is presented: the exact solution by brute force, the classical approximation based on the Goemans-Williamson (GW) algorithm, the quantum variational approach through the Variational Quantum Eigensolver (VQE), and the Quantum Approximate Optimization Algorithm (QAOA). To validate the feasibility of these methodologies in practical scenarios, this work incorporates a case study focused on the structural optimization of a telecommunications network where the algorithms were applied. The results demonstrate that the classical Goemans-Williamson algorithm offers the best cost-benefit, maintaining high temporal efficiency and an AR close to 1,0000. Among the quantum models, a *trade-off* became evident: the VQE presented higher solution quality at the expense of a parametric space of order $\mathcal{O}(N)$, while the QAOA achieved excellent parametric compression in the order of $\mathcal{O}(p)$, but with degradation in the cut quality as the problem scaled. It is concluded that the main current barrier to quantum advantage in these formulations does not lie in the mathematics of the *ansätze*, but in the exponential cost of the classical simulation of state vectors, reinforcing the need for execution on real quantum hardware to obtain practical gains.

Keywords: MaxCut. Graphs. VQE. Goemans-Williamson Algorithm. Quantum Computing. Qiskit.

LISTA DE FIGURAS

Figura 1 – Um grafo não orientado G	17
Figura 2 – Representação de um grafo por meio de listas de adjacências.	19
Figura 3 – Representação matricial de um grafo.	19
Figura 4 – Representação visual do problema MaxCut em um grafo	20
Figura 5 – Arquitetura de hardware para algoritmos híbridos na era <i>Noisy Intermediate-Scale Quantum</i> - NISQ	25
Figura 6 – Exemplo de circuito para o QAOA (painel A) e o correspondente Grafo de Precedência Lógica dos Dados (painel B). O painel C exibe o grafo de interação para profundidade $p = 2$ e o grafo de conectividade do hardware. A otimização do escalonamento permite que as portas que compõem o operador $U_C(\gamma)$ sejam executadas em paralelo, reduzindo a profundidade efetiva do circuito.	29
Figura 7 – Hierarquia de complexidade assintótica	34
Figura 8 – Fluxograma: Comparação de Métodos Max-Cut.	40
Figura 9 – Razão de aproximação média dos métodos avaliados em função do tamanho do grafo.	43
Figura 10 – Crescimento da dimensionalidade paramétrica dos métodos avaliados em escala logarítmica.	45
Figura 11 – Tempo de execução estimado dos métodos avaliados em função do tamanho do grafo (escala log).	46
Figura 12 – Diagrama original da topologia de <i>backbone</i> da NSFNET (1991).	48
Figura 13 – Reconstrução e modelagem em grafo da topologia NSFNET.	48

LISTA DE TABELAS

Tabela 1	–	Resultado da média e desvio-padrão ($\mu \pm \sigma$) para ($N = 14$ a 20).	47
Tabela 2	–	Resultado da média e desvio-padrão ($\mu \pm \sigma$) para o estudo de caso NSFNET.	49

LISTA DE ABREVIATURAS E SIGLAS

AR	<i>Approximation Ratio</i>
CPU	<i>Central Processing Unit</i>
CVXPY	biblioteca de programação convexa em Python
GW	<i>Goemans-Williamson algorithm</i>
HDD	<i>Hard Disk Drive</i>
HEA	<i>Hardware-Efficient Ansätze</i>
MaxCut	<i>Maximum Cut</i>
MPS	<i>Matrix Product State</i>
NISQ	<i>Noisy Intermediate-Scale Quantum</i>
NP	<i>Nondeterministic Polynomial time</i>
P	<i>Polynomial time</i>
PSD	<i>Positive Semidefinite</i>
QAOA	<i>Quantum Approximate Optimization Algorithm</i>
QCU	Quantum Control Unit
QEU	Quantum Processing Unit
Qiskit	kit de desenvolvimento de software para computação quântica
QPU	<i>Quantum Processing Unit</i>
QPUs	<i>Unidades de Processamento Quântico</i>
RAM	<i>Random Access Memory</i>
SCS	<i>Splitting Conic Solver</i>
SDP	<i>Semidefinite Programming</i>
VQAs	<i>Variational Quantum Algorithms</i>
VQE	<i>Variational Quantum Eigensolver</i>

LISTA DE SÍMBOLOS

G	Grafo não orientado
V	Conjunto de vértices (nós)
E	Conjunto de arestas (enlaces)
N	Número total de vértices
w_{ij}	Peso da aresta entre os vértices i e j
$\mathcal{O}$	Notação assintótica de complexidade (Big-O)
$\hat{H}$	Operador Hamiltoniano
Z	Operador quântico de Pauli Z
X	Operador quântico de Pauli X
$ \psi\rangle$	Vetor de estado quântico
$\vec{\theta}$	Vetor de parâmetros clássicos do circuito
p	Profundidade do circuito quântico (camadas)
AR	Razão de Aproximação (<i>Approximation Ratio</i>)

SUMÁRIO

1	INTRODUÇÃO	14
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Grafos e Teoria Básica de Otimização Combinatória	17
2.1.1	<i>Definições Fundamentais</i>	17
2.1.2	<i>Representação Computacional de Grafos</i>	18
2.1.3	<i>Otimização Combinatória em Grafos</i>	19
2.2	O Problema do MaxCut	20
2.2.1	<i>Formulações Matemáticas do MaxCut</i>	20
2.3	Complexidade Computacional e NP-dificuldade	21
2.4	Programação Semidefinida	22
2.5	O Algoritmo de Goemans-Williamson	22
2.6	Computação Quântica na Era NISQ e Algoritmos Variacionais	24
2.6.1	<i>Fundamentação Matemática e o Loop Híbrido</i>	24
2.6.2	<i>Estrutura do Ansatz e Circuitos Quânticos Parametrizados</i>	25
2.6.3	<i>O Algoritmo Variational Quantum Eigensolver (VQE) aplicado ao MaxCut</i>	26
2.6.4	<i>Quantum Approximate Optimization Algorithm</i>	27
2.7	Relação com a Complexidade e Benchmarking Clássico	29
2.8	Ferramentas Computacionais	30
2.9	Métricas de Avaliação e Modelagem Estatística	31
2.9.1	<i>Razão de Aproximação</i>	31
2.9.2	<i>Simulação Estocástica de Monte Carlo</i>	32
2.9.3	<i>Dimensionalidade e Expressividade</i>	32
2.9.4	<i>Projeção Empírica via Regressão Analítica</i>	34
3	METODOLOGIA	36
3.1	Caracterização do Método de Pesquisa	36
3.2	Modelagem das Instâncias de Rede	36
3.3	Escopo Experimental e Amostragem (Método de Monte Carlo)	36
3.4	Configuração dos Algoritmos Solucionadores	37
3.5	Coleta de Métricas e Regressão Preditiva	38
3.5.1	<i>Infraestrutura Computacional</i>	38

4	IMPLEMENTAÇÃO COMPUTACIONAL	39
4.1	Arquitetura do Código	39
4.2	Pré-processamento dos Grafos	39
4.3	Implementação da Força Bruta	39
4.4	Implementação do Goemans-Williamson	41
4.5	Implementação do VQE e QAOA	41
4.6	Organização dos Resultados	42
4.7	Geração de Gráficos e Tabelas	42
4.8	Validação dos Experimentos	42
5	RESULTADOS E DISCUSSÃO	43
5.1	Qualidade da Solução e Razão de Aproximação	43
5.2	Dimensionalidade Paramétrica	44
5.3	Projeção de Escalabilidade Temporal	45
5.4	Síntese dos Resultados	46
5.5	Estudo de Caso: Topologia NSFNET (1991)	47
6	CONCLUSÕES E TRABALHOS FUTUROS	50
	REFERÊNCIAS BIBLIOGRÁFICAS	52
	APÊNDICE A – Geração de Instâncias	56
	APÊNDICE B – Implementação da Força Bruta	58
	APÊNDICE C – Implementação do Goemans-Williamson	59
	APÊNDICE D – Implementação do VQE e QAOA	61

1 INTRODUÇÃO

Em redes de telecomunicações, diversos problemas de alocação e particionamento podem ser modelados como tarefas de otimização combinatória, nas quais é necessário equilibrar desempenho, custo computacional e escalabilidade (RESENDE; PARDALOS, 2006). Entre esses problemas, o *Maximum Cut* (MaxCut) ocupa posição de destaque por sua relação com particionamento de grafos, balanceamento de carga, segmentação de redes e outras aplicações em infraestrutura de comunicação.

O problema MaxCut consiste em particionar o conjunto de vértices de um grafo em dois subconjuntos de modo a maximizar o número, ou o peso, das arestas que atravessam essa divisão. Trata-se de um problema NP-difícil (KARP, 1972; GAREY; JOHNSON, 1979), de modo que algoritmos exatos apresentam crescimento exponencial e se tornam inviáveis para instâncias maiores. Em razão disso, abordagens aproximadas e heurísticas desempenham papel central na prática, especialmente quando há necessidade de respostas eficientes em cenários de maior escala (VAZIRANI, 2002).

Entre os métodos clássicos mais relevantes para MaxCut está o algoritmo de Goemans-Williamson (GW), baseado em relaxação por programação semidefinida (*Semidefinite Programming* - SDP) e arredondamento aleatório (GOEMANS; WILLIAMSON, 1995). Esse método é amplamente reconhecido por sua garantia de aproximação de 0,8785 e permanece como referência clássica para o problema (WILLIAMSON; SHMOYS, 2011). Sua importância não está apenas no desempenho prático, mas também no papel que desempenha como limite de comparação para novas abordagens de otimização.

Com o avanço da computação quântica, passaram a ser investigados algoritmos híbridos capazes de explorar recursos quânticos e clássicos. Nesse contexto, métodos variacionais como o *Variational Quantum Eigensolver* (VQE) e o *Quantum Approximate Optimization Algorithm* (QAOA) são estudados para a resolução de problemas combinatórios, incluindo formulações equivalentes ao MaxCut por meio do mapeamento para o modelo de Ising (FARHI *et al.*, 2014; FARHI *et al.*, 2017; KANDALA *et al.*, 2017; WECKER *et al.*, 2016; CEREZO *et al.*, 2021; LUCAS, 2014). Em particular, essas abordagens oferecem estruturas flexíveis para representar soluções aproximadas por meio de circuitos, embora o desempenho prático dependa fortemente da escolha do *ansatz*, da eficiência do otimizador clássico e do custo inerente à simulação ou execução em hardware quântico (CEREZO *et al.*, 2021; KANDALA *et al.*, 2017).

Diante desse cenário, este trabalho apresenta um *benchmarking* abrangente entre

abordagens clássicas e quânticas para o problema MaxCut: a solução exata por Força Bruta, a aproximação clássica algoritmo de GW, e as metodologias variacionais quânticas por meio do VQE e do QAOA. As instâncias do problema foram modeladas como grafos aleatórios com topologia controlada e avaliadas em um estudo de caso prático focado na topologia da rede NSFNET. O desempenho dos algoritmos foi avaliado em termos de Razão de Aproximação, escalabilidade temporal e dimensionalidade paramétrica. A principal contribuição deste estudo consiste em comparar, sob as mesmas condições de estresse combinatorial, o comportamento de um método exato, de um método clássico consolidado e de algoritmos híbridos quânticos aplicados a topologias inspiradas em redes de telecomunicações reais.

O trabalho está organizado da seguinte forma:

- **Capítulo 2 – Fundamentação Teórica:** Apresenta a teoria básica de otimização combinatorial em grafos, a formulação do problema MaxCut e sua complexidade computacional. Detalha as abordagens clássicas, com foco na Programação Semidefinida e no algoritmo de Goemans-Williamson. Introduz a computação quântica na era NISQ, descrevendo os algoritmos variacionais (VQE e QAOA), o mapeamento para o Hamiltoniano de Ising e as ferramentas e métricas estatísticas utilizadas na avaliação.
- **Capítulo 3 – Metodologia:** Caracteriza o método de pesquisa experimental baseado em simulação computacional. Detalha a modelagem das instâncias de rede, a aplicação do método de Monte Carlo no escopo de amostragem, a configuração paramétrica dos algoritmos solucionadores e a infraestrutura computacional exigida para a coleta empírica de métricas.
- **Capítulo 4 – Implementação Computacional:** Expõe a arquitetura do código desenvolvido e as etapas de pré-processamento dos grafos. Apresenta os detalhes práticos da codificação dos métodos abordados (Força Bruta, Goemans-Williamson, VQE e QAOA), além dos processos de organização de dados, geração visual de resultados e validação dos experimentos.
- **Capítulo 5 – Resultados e Discussão:** Dedicar-se à análise comparativa e crítica das metodologias avaliadas. Avalia sistematicamente a qualidade das soluções (Razão de Aproximação), a evolução da dimensionalidade paramétrica e as projeções de escalabilidade temporal, consolidando uma síntese dos limites e viabilidades de cada método, aplicados ao estudo de caso topologia NSFNET.
- **Capítulo 6 – Conclusões e Trabalhos Futuros:** Sintetiza os achados centrais da pesquisa,

avaliando o comportamento das abordagens clássicas e quânticas frente ao problema proposto. Aponta as atuais limitações da simulação quântica em hardware clássico e sugere direcionamentos práticos para estudos futuros, como a execução em (*Unidades de Processamento Quântico* - QPUs) reais.

2 FUNDAMENTAÇÃO TEÓRICA

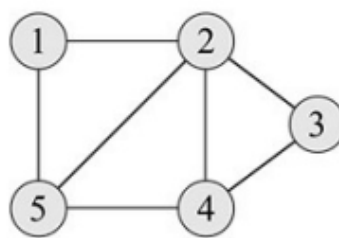
2.1 Grafos e Teoria Básica de Otimização Combinatória

A modelagem de problemas em redes de telecomunicações exige uma infraestrutura matemática complexa, uma vez que diversos problemas de alocação e particionamento podem ser modelados como tarefas de otimização combinatória, nas quais é necessário equilibrar desempenho, custo computacional e escalabilidade (RESENDE; PARDALOS, 2006). A teoria de grafos e a otimização combinatória fornecem a linguagem e as ferramentas analíticas para representar, analisar e resolver problemas de alocação, roteamento e particionamento.

2.1.1 Definições Fundamentais

Formalmente, um grafo é definido como uma tripla estrutural, embora seja frequentemente simplificado como o par $G = (V, E)$, onde V é um conjunto finito de elementos chamados de vértices (ou nós), e E é um conjunto de pares não ordenados de vértices, denominados arestas (DIESTEL, 2017). A Figura 1 ilustra um exemplo de grafo não orientado, mostrando a relação entre os vértices e suas conexões. Quando as relações possuem uma direção específica, define-se um grafo orientado (ou dígrafo), no qual as arestas são pares ordenados (u, v) , indicando que a aresta tem origem no vértice u e destino no vértice v .

Figura 1 – Um grafo não orientado G .



Fonte: Adaptado de (CORMEN *et al.*, 2009, p. 485).

Assim, dois vértices $u, v \in V$ são adjacentes (ou vizinhos) se existe uma aresta os conectando. Nesse caso, diz-se que a aresta é conectada a esses vértices. Grafos que não possuem arestas paralelas (múltiplas arestas conectando o mesmo par de vértices) ou laços (arestas conectando um vértice a si mesmo) são chamados de grafos simples (KORTE; VYGEN, 2012; DIESTEL, 2017).

Em muitos problemas de otimização, analisa-se apenas uma parte da rede. Um

grafo $H = (V_H, E_H)$ é um subgrafo de $G = (V, E)$ se $V_H \subseteq V$ e $E_H \subseteq E$. Caso H contenha todas as arestas de G conectadas aos vértices em V_H , diz-se que H é um subgrafo induzido por V_H (DIESTEL, 2017).

Uma métrica estrutural para a caracterização de uma topologia é a sua densidade (D). A densidade mede a proporção de arestas existentes no grafo em relação ao número máximo de arestas possíveis. Para um grafo simples e não orientado contendo $|V|$ vértices (frequentemente denotado como N) e $|E|$ arestas, o número máximo de conexões possíveis é dado por $|V|(|V| - 1)/2$. Dessa forma, a densidade é definida matematicamente pela razão (DIESTEL, 2017):

$$D = \frac{2|E|}{|V|(|V| - 1)}, \quad (1)$$

onde $0 \leq D \leq 1$. Grafos que possuem um número de conexões muito inferior ao máximo possível (D tendendo a 0) são classificados como grafos esparsos. Em contrapartida, grafos com alta conectividade (D tendendo a 1) são classificados como densos (KORTE; VYGEN, 2012). Essa métrica é crucial não apenas para analisar a resiliência da rede, mas também para determinar a estrutura de dados mais eficiente para a sua representação algorítmica (CORMEN *et al.*, 2009).

2.1.2 Representação Computacional de Grafos

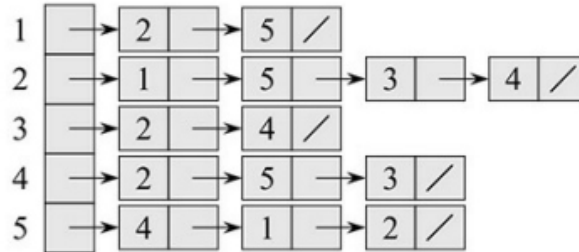
Para que os algoritmos de combinatória operem sobre essas estruturas, é necessário mapeá-las para a memória do computador. Existem duas abordagens para representar um grafo $G = (V, E)$: listas de adjacências e matrizes de adjacências (CORMEN *et al.*, 2009). A escolha entre elas depende da topologia do grafo e da natureza do algoritmo a ser aplicado.

A **representação por lista de adjacências** consiste em um arranjo Adj contendo $|V|$ listas, uma para cada vértice. Para cada $u \in V$, a lista $Adj[u]$ contém todos os vértices v adjacentes a u . Esta abordagem é eficiente para representar grafos esparsos — onde o número de arestas $|E|$ é muito menor que $|V|^2$ —, pois a quantidade de memória exigida é proporcional a $\Theta(V + E)$ (CORMEN *et al.*, 2009). A Figura 2 exemplifica essa organização, detalhando como o arranjo Adj aponta para as listas de vértices adjacentes.

Por outro lado, a **representação por matriz de adjacências** assume que os vértices são numerados de 1 a $|V|$, formando uma matriz A de dimensões $|V| \times |V|$. A entrada a_{ij} recebe o valor 1 (ou o peso da aresta) se existe uma conexão entre os vértices i e j , e 0 (ou nulo) caso contrário. Em grafos não orientados, essa matriz é estritamente simétrica em relação à sua

diagonal principal ($A = A^T$). Essa estrutura de linhas e colunas, permite identificar a existência de conexões, é ilustrada na Figura 3.

Figura 2 – Representação de um grafo por meio de listas de adjacências.



Fonte: Adaptado de (CORMEN *et al.*, 2009, p. 485).

Embora exija uma complexidade espacial de $\Theta(V^2)$ independentemente da quantidade de arestas, a matriz de adjacências é preferível para grafos densos (onde $|E|$ se aproxima de $|V|^2$) ou quando o algoritmo exige consultas instantâneas em tempo constante $O(1)$ para verificar a existência de uma aresta específica (CORMEN *et al.*, 2009).

Figura 3 – Representação matricial de um grafo.

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

Fonte: Adaptado de (CORMEN *et al.*, 2009, p. 485).

2.1.3 Otimização Combinatória em Grafos

A teoria de grafos é a base onde a otimização combinatória atua. O objetivo central é encontrar um subconjunto de vértices ou arestas que minimize ou maximize uma função de custo específica. Para isso, os grafos são modelados como grafos ponderados, onde uma função $w : E \rightarrow \mathbb{R}$ atribui pesos (representando custos ou distâncias) a cada aresta (KORTE; VYGEN, 2012).

O grande desafio dessa área reside na explosão combinatória. Em problemas como o (MaxCut), o espaço de soluções cresce exponencialmente com o aumento do número de vértices. Uma vez que a varredura de todas as combinações (força bruta) torna-se computacionalmente

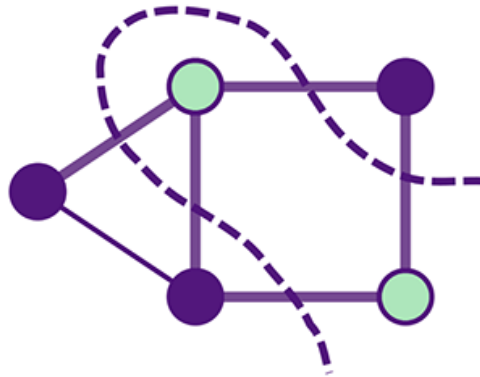
inviável, é necessário o desenvolvimento de estratégias heurísticas, aproximações clássicas e, mais recentemente, métodos híbridos quânticos para realizar a busca de forma eficiente.

2.2 O Problema do MaxCut

Dentre os modelos de grafos amplamente utilizados, o problema do *Maximum Cut* (MaxCut) ocupa uma posição de destaque, com vastas aplicações que se estendem desde o *layout* de circuitos físicos até a modelagem de sistemas em física estatística (BARAHONA *et al.*, 1988).

Dessa forma, considere um grafo não orientado $G = (V, E)$, onde V é o conjunto de vértices, com $|V| = N$, e E é o conjunto de arestas. O problema do MaxCut consiste em particionar o conjunto V em dois subconjuntos disjuntos, S e $V \setminus S$, de modo a maximizar o número das arestas que atravessam essa partição. A Figura 4 mostra essa divisão em um grafo, onde as arestas cortadas pelos subconjuntos são destacadas, mostrando o objetivo de maximização da capacidade de corte.

Figura 4 – Representação visual do problema MaxCut.



Fonte: Adaptado de Intellilink (2026). Disponível em:
<https://www.intellilink.co.jp/en/business/software/column2023102600.aspx>.
 Acesso em: 22 abr. 2026.

2.2.1 Formulações Matemáticas do MaxCut

Para formular esse problema algebricamente e permitir sua análise computacional, existem duas abordagens principais de mapeamento de variáveis.

A primeira abordagem associa a cada vértice $i \in V$ uma variável binária de *spin* $z_i \in \{-1, 1\}$. Assim, $z_i = 1$ indica que o vértice pertence ao subconjunto S , enquanto $z_i = -1$ indica que ele pertence a $V \setminus S$. O custo do corte é dado pela maximização da seguinte função

com variáveis quadráticas não restritas:

$$C(z) = \frac{1}{2} \sum_{(i,j) \in E} w_{ij}(1 - z_i z_j), \quad (2)$$

onde w_{ij} representa o peso da aresta conectando os vértices i e j . O termo $(1 - z_i z_j)$ resulta em 2 se os vértices estiverem em conjuntos distintos e 0 se estiverem no mesmo conjunto.

A segunda abordagem, demonstrada por Deza e Laurent (DEZA; LAURENT, 1997), utiliza um vetor de decisão puramente binário $x \in \{0, 1\}^N$, onde $x_i = 1$ se $i \in S$ e $x_i = 0$ se $i \in V \setminus S$. Seja $W = [w_{ij}]_{i,j=1}^N$ a matriz de adjacência ponderada do grafo com diagonal nula ($w_{ii} = 0$), e $e = [1, 1, \dots, 1]^T$ o vetor unitário. O problema pode ser reescrito como a otimização de uma função quadrática sobre um hipercubo:

$$\max_{x \in \{0,1\}^N} x^T W (e - x). \quad (3)$$

Como a matriz W possui diagonal nula, a função objetivo na Equação 3 é linear em relação a cada variável individualmente, o que possibilita a transição do domínio discreto para relaxações contínuas no hipercubo $x \in [0, 1]^N$.

Portanto, para determinar a solução exata por força bruta, é necessário avaliar todas as $2^{(N-1)}$ combinações possíveis de partição. Em decorrência de sua natureza NP-difícil (KARP, 1972; GAREY; JOHNSON, 1979), o custo computacional de $\mathcal{O}(2^N)$ apresenta crescimento exponencial, tornando a abordagem exaustiva inviável para instâncias de larga escala.

2.3 Complexidade Computacional e NP-dificuldade

A teoria da Complexidade Computacional dedica-se a classificar problemas matemáticos e algoritmos com base na quantidade de recursos consumidos, como o tempo de execução e espaço de memória, necessários para resolvê-los em função da dimensão de sua entrada (TOSCANI; VELOSO, 2001; SIPSER, 2012). O objetivo central dessa análise é estabelecer limites teóricos sobre o que pode ser computado de maneira eficiente, mapeando problemas tratáveis e intratáveis (ARORA; BARAK, 2009).

Assim, os problemas são agrupados em classes de complexidade. A classe *Polynomial time* - P contém os problemas que podem ser resolvidos eficientemente, ou seja, em tempo polinomial por uma máquina de Turing determinística (MTD). Por outro lado, a classe NP (Nondeterministic Polynomial time) abrange os problemas cujas soluções podem ser verificadas como corretas em tempo polinomial por uma MTD, independentemente do esforço necessário para encontrá-las inicialmente (TOSCANI; VELOSO, 2001; CORMEN *et al.*, 2009).

A análise de problemas de otimização combinatória reside na classe dos problemas NP-difíceis (*NP-hard*). Segundo o trabalho seminal de Garey e Johnson (GAREY; JOHNSON, 1979), um problema é classificado como NP-difícil se ele for pelo menos tão complexo quanto os problemas mais difíceis da classe NP. A implicação prática dessa propriedade é profunda: a descoberta de um algoritmo exato e de tempo polinomial para um único problema NP-difícil implicaria que $P = NP$, resolvendo a questão em aberto da ciência da computação teórica (PAPADIMITRIOU, 2003).

2.4 Programação Semidefinida

A programação semidefinida (SDP) é uma extensão da programação linear que permite otimizar funções sobre o cone das matrizes simétricas semidefinidas. De acordo com Vandenberghe e Boyd (VANDENBERGHE; BOYD, 1996), a SDP oferece uma base poderosa para a resolução de problemas de otimização combinatória, permitindo a construção de relações convexas.

Nesse contexto, o interesse pela SDP cresceu significativamente devido à sua capacidade de aproximar problemas discretos com garantias de desempenho teóricas (BOYD; VANDENBERGHE, 2004). Como discutido por Wolkowicz et al. (WOLKOWICZ *et al.*, 2012), problemas que, na sua forma original, apresentam um espaço de busca combinatório (como o MaxCut), podem ser mapeados para o domínio contínuo de matrizes, onde técnicas de pontos interiores garantem convergência em tempo polinomial (NESTEROV; NEMIROVSKII, 1994). Além disso, a flexibilidade da SDP é ampliada pelo uso de bibliotecas de modelagem como biblioteca de programação convexa em Python - CVXPY, que facilitam a implementação computacional de problemas complexos (DIAMOND; BOYD, 2016).

2.5 O Algoritmo de Goemans-Williamson (GW)

O algoritmo de GW, introduzido por Goemans e Williamson (GOEMANS; WILLIAMSON, 1995), representa um marco na teoria de algoritmos de aproximação. Ao aplicar a relaxação via SDP ao MaxCut, os autores demonstraram ser possível obter um fator de aproximação (α) de 0,8785 em relação ao valor exato, superando as abordagens clássicas baseadas em algoritmos gulosos (*greedy algorithms*) ou busca local (WILLIAMSON; SHMOYS, 2011).

Formalmente, a modelagem desse relaxamento consiste em estender as variáveis

discretas de spin $z_i \in \{-1, 1\}$ para vetores unitários $v_i \in \mathbb{R}^N$ dispostos sobre uma hipersfera unitária (GOEMANS; WILLIAMSON, 1995). Para o problema ser formulado de maneira convexa, define-se uma matriz simétrica X cujas entradas correspondem ao produto escalar desses vetores, isto é, $X_{ij} = v_i \cdot v_j$. Dessa forma, o problema de otimização combinatória é transformado em um programa semidefinido convexo de maximização, expresso por (WILLIAMSON; SHMOYS, 2011):

$$\text{maximizar } \frac{1}{2} \sum_{(i,j) \in E} w_{ij}(1 - X_{ij}). \quad (4)$$

O sucesso do método reside na transformação dos vértices em vetores sobre uma hipersfera unitária, um processo conhecido como *randomized rounding* (MOTWANI; RAGHAVAN, 1996). Esta técnica de arredondamento transforma a solução contínua da SDP em uma partição discreta válida para o MaxCut. Conforme apresentado por Williamson e Shmoys (WILLIAMSON; SHMOYS, 2011), a análise geométrica do hiperplano de separação é o que garante esse desempenho.

Numericamente, o fator de aproximação $\alpha \approx 0,87856$ é definido pela pior razão possível entre a probabilidade de um hiperplano aleatório separar dois vetores v_i e v_j e a contribuição correspondente desses mesmos vetores na função objetivo do relaxamento semidefinido, dada por $\frac{1}{2}(1 - \cos \theta)$ (GOEMANS; WILLIAMSON, 1995). O limite inferior é obtido resolvendo o seguinte problema de minimização (GOEMANS; WILLIAMSON, 1995; WILLIAMSON; SHMOYS, 2011):

$$\alpha = \min_{\theta \in [0, \pi]} \frac{2\theta}{\pi(1 - \cos \theta)}. \quad (5)$$

Nesta formulação, α representa o limite inferior de desempenho do algoritmo, garantindo matematicamente que o custo do corte encontrado será, no pior dos casos, pelo menos 87,85% (0,87856) do valor do corte máximo. Esse valor mínimo global ocorre quando o ângulo θ entre os vetores na hipersfera é de aproximadamente 2,3311 radianos. Esta garantia teórica consolidou o método de Goemans-Williamson como a referência para o *benchmarking* de novos paradigmas computacionais, estabelecendo o limite clássico que os algoritmos variacionais em computação quântica buscam superar (HÅSTAD, 2001; LUENBERGER *et al.*, 1984).

2.6 Computação Quântica na Era NISQ e Algoritmos Variacionais

A era da computação quântica conhecida como Noisy Intermediate-Scale Quantum (NISQ) abriu novas possibilidades para a abordagem de problemas de otimização combinatória, como o MaxCut (KARP, 1972). Diferentemente dos futuros computadores quânticos tolerantes a falhas, os dispositivos NISQ possuem um número limitado de qubits (dezenas a centenas) e sofrem com ruídos, além de tempos de coerência curtos (PRESKILL, 2018). Devido a essas severas restrições de hardware, a execução de algoritmos quânticos de longa profundidade torna-se inviável na prática. Nesse cenário, a abordagem mais favorável reside nos Algoritmos Quânticos Variacionais (*Variational Quantum Algorithms* - VQAs), que utilizam uma arquitetura híbrida quântica-clássica para minimizar a profundidade dos circuitos quânticos (CEREZO *et al.*, 2021; MCCLEAN *et al.*, 2016a).

2.6.1 Fundamentação Matemática e o Loop Híbrido

O princípio fundamental dos algoritmos variacionais quânticos baseia-se em utilizar uma Unidade de Processamento Quântico (*Quantum Processing Unit* - QPU) para executar a tarefa de preparação e medição de um estado quântico parametrizado, enquanto um otimizador clássico (*Central Processing Unit* - CPU) ajusta iterativamente esses parâmetros. O base matemática dessa abordagem é o princípio variacional de Rayleigh-Ritz da mecânica quântica (NIELSEN; CHUANG, 2010).

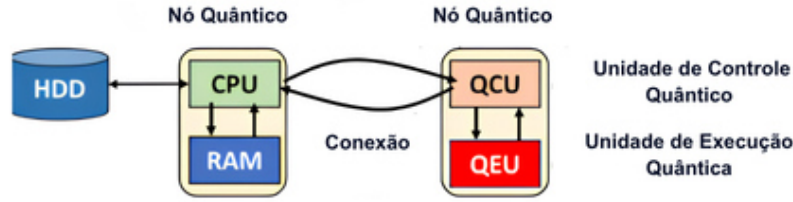
Dado um operador $\hat{H}$ cujos autovalores são limitados inferiormente por uma energia fundamental E_0 , o valor esperado da energia para qualquer estado quântico de tentativa parametrizado $|\psi(\vec{\theta})\rangle$ será sempre maior ou igual a E_0 :

$$E(\vec{\theta}) = \frac{\langle \psi(\vec{\theta}) | \hat{H} | \psi(\vec{\theta}) \rangle}{\langle \psi(\vec{\theta}) | \psi(\vec{\theta}) \rangle} \geq E_0. \quad (6)$$

Assumindo a normalização do estado ($\langle \psi | \psi \rangle = 1$), a execução algorítmica exige uma infraestrutura de hardware fisicamente distribuída.

Como ilustrado na Figura 5, a unidade de processamento clássica composta por CPU, *Random Access Memory* - RAM e *Hard Disk Drive* - HDD comunica-se via rede com a QPU, que atua como um coprocessador. Internamente, a QPU recebe as instruções por meio de uma Unidade de Controle Quântico (*Quantum Control Unit* - QCU), responsável por preparar e medir o estado físico na Unidade de Execução Quântica (*Quantum Processing Unit* - QEU).

Figura 5 – Arquitetura de hardware para algoritmos híbridos na era NISQ.



Fonte: Adaptado de Rauch, Rontani e Vialle (2025).

Após a QPU avaliar a função de custo e devolver o resultado, o objetivo de resolver o problema de minimização recai sobre a CPU clássica (PERUZZO *et al.*, 2014):

$$\min_{\vec{\theta}} E(\vec{\theta}) = \min_{\vec{\theta}} \langle \psi(\vec{\theta}) | \hat{H} | \psi(\vec{\theta}) \rangle. \quad (7)$$

Para que a avaliação de $E(\vec{\theta})$ seja computável na QEU, o Hamiltoniano $\hat{H}$ é decomposto em uma combinação linear de produtos tensoriais de matrizes de Pauli (formando as *Pauli strings*, denotadas por $P_i \in \{I, X, Y, Z\}^{\otimes N}$): $H = \sum_i c_i P_i$. Pela linearidade da mecânica quântica, o valor esperado total processado é a soma das medições ponderadas pelos coeficientes clássicos c_i (MCCLEAN *et al.*, 2016b):

$$\langle \psi(\vec{\theta}) | \hat{H} | \psi(\vec{\theta}) \rangle = \sum_i c_i \langle \psi(\vec{\theta}) | P_i | \psi(\vec{\theta}) \rangle. \quad (8)$$

2.6.2 Estrutura do Ansatz e Circuitos Quânticos Parametrizados

O *ansatz* é o elemento central dos algoritmos quânticos variacionais, atuando como o circuito quântico parametrizado responsável por preparar o estado de tentativa $|\psi(\vec{\theta})\rangle$. Matematicamente, ele aplica um operador unitário $U(\vec{\theta})$ sobre um estado de referência inicial (tipicamente $|0\rangle^{\otimes N}$), definindo o subespaço de busca que será explorado pelo otimizador clássico (CEREZO *et al.*, 2021; KANDALA *et al.*, 2017).

Na era NISQ, a concepção do *ansatz* exige um equilíbrio entre a sua expressividade e a profundidade do circuito. Circuitos excessivamente profundos acumulam taxas de erro no hardware e sofrem rápida decoerência (PRESKILL, 2018). Adicionalmente, arquiteturas complexas ou sem restrições estruturais correm o risco de enfrentar (*barren plateaus*)¹, o que inviabiliza o treinamento dos parâmetros (MCCLEAN *et al.*, 2018).

¹ O fenômeno dos *barren plateaus* ocorre quando o gradiente da função de custo em algoritmos variacionais torna-se exponencialmente pequeno em relação ao número de qubits, tornando o treinamento do *ansatz* numericamente instável e ineficiente (LAROCCA *et al.*, 2025).

2.6.3 O Algoritmo Variational Quantum Eigensolver (VQE) aplicado ao MaxCut

O algoritmo VQE foi originalmente proposto para determinar a energia do estado fundamental de sistemas moleculares no âmbito da química (PERUZZO *et al.*, 2014). Contudo, sua aplicação estendeu-se com sucesso para a otimização combinatória ao realizar o mapeamento da função de custo de problemas clássicos para o operador Hamiltoniano de um sistema de spins (LUCAS, 2014).

Para a resolução do problema do MaxCut via VQE, é necessário transformar um problema de partição de vértices em um problema de minimização de energia de um Hamiltoniano. A formulação de spin-1/2 do modelo de Ising, que descreve a interação magnética entre spins em uma rede estruturada, demonstra-se ideal para essa finalidade (BARAHONA *et al.*, 1988).

Considere a função de custo clássica do MaxCut definida sobre variáveis de spin discretas $z_i \in \{-1, 1\}$, em que $z_i = 1$ indica que o vértice pertence ao subconjunto S e $z_i = -1$ indica que pertence a $V \setminus S$, conforme expresso na Equação (2) (FARHI *et al.*, 2014; LUCAS, 2014):

$$C(z) = \frac{1}{2} \sum_{(i,j) \in E} w_{ij}(1 - z_i z_j). \quad (9)$$

A maximização do peso total das arestas atravessam o corte é matematicamente equivalente à minimização do termo de spins $\sum_{(i,j) \in E} w_{ij} z_i z_j$ (FARHI *et al.*, 2014; LUCAS, 2014). No paradigma da computação quântica, as variáveis clássicas de spin z_i são mapeadas diretamente para os operadores quânticos de Pauli Z, denotados por Z_i , os quais atuam sobre o qubit i . A matriz correspondente ao operador Pauli Z assume a seguinte forma na base computacional $\{|0\rangle, |1\rangle\}$ (NIELSEN; CHUANG, 2010):

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (10)$$

Os autovalores associados a Z são 1 e -1 , correspondendo, respectivamente, aos estados próprios $|0\rangle$ (onde $z_i = 1$) e $|1\rangle$ (onde $z_i = -1$) (NIELSEN; CHUANG, 2010). Desse modo, o mapeamento entre o domínio clássico e o espaço de estados quânticos estabelece-se de forma direta, a variável clássica $z_i = 1$ ao estado $|0\rangle_i$ e $z_i = -1$ ao estado quântico $|1\rangle_i$.

Ao substituir as variáveis discretas clássicas pelos operadores de Pauli, obtém-se o Hamiltoniano de Custo (H_C) associado ao problema do MaxCut (FARHI *et al.*, 2014; MCCLEAN

et al., 2016b):

$$\hat{H}_C = \frac{1}{2} \sum_{(i,j) \in E} w_{ij} Z_i Z_j, \quad (11)$$

onde I denota a matriz identidade. O propósito do VQE aplicado ao MaxCut consiste, portanto, em encontrar o estado fundamental (estado de menor energia) desse operador (MCCLEAN *et al.*, 2016a; STECHLY *et al.*, 2019).

A escolha do *ansatz* que representa a forma funcional do circuito quântico parametrizado responsável por gerar o estado $|\psi(\vec{\theta})\rangle$ assume um papel crítico no desempenho do VQE. Na era NISQ, os chamados *ansatze* eficientes em hardware *Hardware-Efficient Ansatz* - HEA são frequentemente usados para minimizar a profundidade do circuito e reduzir os efeitos da decoerência (KANDALA *et al.*, 2017).

Matematicamente, este operador é estruturado em blocos que combinam portas de rotação de um único qubit com camadas de emaranhamento (U_{ent}). O *ansatz* geral com profundidade D é gerado pela aplicação sequencial dessas camadas sobre o estado inicial $|0\rangle^{\otimes N}$ (KANDALA *et al.*, 2017):

$$|\psi(\vec{\theta})\rangle = \prod_{d=1}^D \left(U_{\text{ent}} \bigotimes_{j=1}^N R_y(\theta_{j,d}) R_z(\phi_{j,d}) \right) |0\rangle^{\otimes N}. \quad (12)$$

Nesta equação (12), $\vec{\theta}$ representa o vetor de parâmetros variacionais, composto pelos ângulos $\theta_{j,d}$ e $\phi_{j,d}$, que constroem as variáveis utilizadas pelo otimizador clássico. O índice $j \in \{1, \dots, N\}$ identifica o qubit sobre o qual a porta quântica atua, onde N corresponde ao número total de qubits. O índice $d \in \{1, \dots, D\}$ denota a camada do circuito, sendo D a profundidade do *ansatz*. Os operadores R_y e R_z representam as portas de rotação de um único qubit em torno dos eixos Y e Z, sendo expressos pelas exponenciais matriciais $R_y(\theta_{j,d}) = e^{-i\theta_{j,d}Y/2}$ e $R_z(\phi_{j,d}) = e^{-i\phi_{j,d}Z/2}$, em que Y e Z são as matrizes de Pauli, estando a matriz Z já definida pela Equação (10). O operador U_{ent} atua como a camada de emaranhamento, implementada por meio de portas de dois qubits aplicadas. A aplicação sequencial dessas operações a partir do estado inicial $|0\rangle^{\otimes N}$ garante a expressividade necessária para que o estado gerado $|\psi(\vec{\theta})\rangle$ possa aproximar o estado fundamental do sistema.

2.6.4 Quantum Approximate Optimization Algorithm (QAOA)

O algoritmo QAOA é um algoritmo variacional, criado com o foco em obtenção de soluções aproximadas para problemas de otimização combinatória em grafos (FARHI *et al.*,

2014). Enquanto o VQE emprega uma estrutura de *ansatz* genérica ditada pelas limitações físicas do hardware, o QAOA constrói o seu circuito quântico baseando-se diretamente na estrutura do problema (FARHI *et al.*, 2014).

A fundamentação do método inspira-se no conceito de evolução adiabática quântica. O QAOA gera o estado quântico final por meio da aplicação alternada de dois operadores unitários parametrizados, os quais são construídos a partir de dois Hamiltonianos que não comutam entre si (FARHI *et al.*, 2014):

1. Hamiltoniano de Custo (H_C): Corresponde ao Hamiltoniano de Ising deduzido anteriormente. Seu operador de evolução unitária, parametrizado por γ , altera as fases relativas dos estados na base computacional (FARHI *et al.*, 2014):

$$U_C(\gamma) = e^{-i\gamma H_C} = \prod_{(i,j) \in E} e^{-i\gamma w_{ij}(I - Z_i Z_j)/2}. \quad (13)$$

2. Hamiltoniano de Mistura (H_M): Destina-se a induzir transições quânticas entre os estados, permitindo a exploração do espaço de soluções. Adota-se a soma dos operadores Pauli X aplicados a todos os qubits: $H_M = \sum_{j=1}^N X_j$. O operador de evolução é (FARHI *et al.*, 2014):

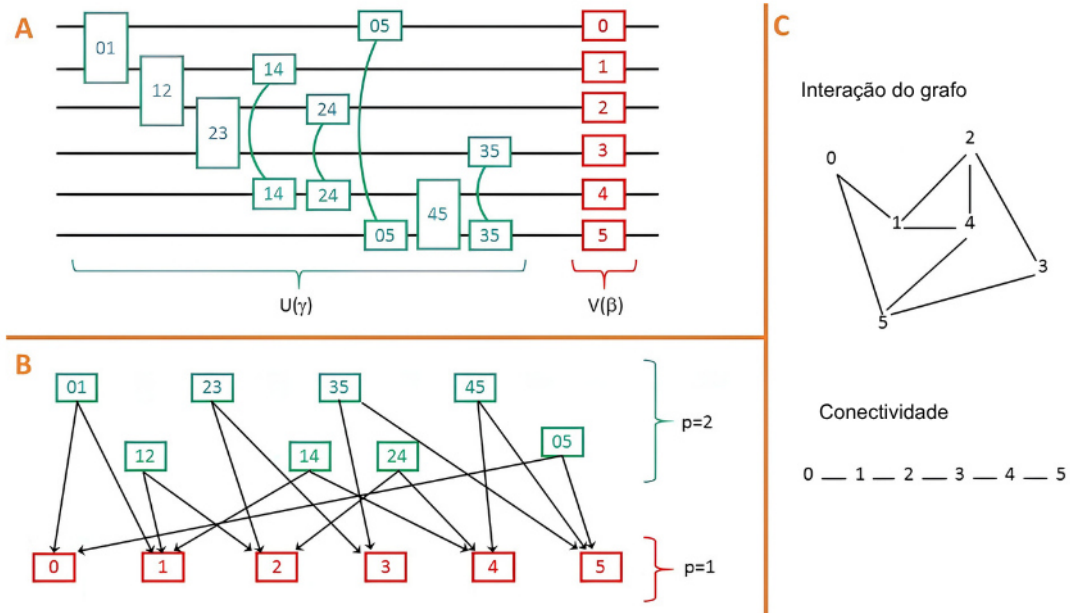
$$U_M(\beta) = e^{-i\beta H_M} = \prod_{j=1}^N e^{-i\beta X_j}. \quad (14)$$

O *ansatz* característico do QAOA com profundidade p é construído inicializando o sistema em uma superposição uniforme de todos os estados possíveis $|s\rangle = |+\rangle^{\otimes N}$, seguida pela aplicação de p camadas de operadores alternados, controlados pelos vetores de parâmetros clássicos $\vec{\gamma}$ e $\vec{\beta}$ (FARHI *et al.*, 2014). Conforme ilustrado na Figura 6, a estrutura do circuito reflete a intercalação dos operadores, onde a organização lógica das portas permite otimizar na execução do operador de custo $U_C(\gamma)$:

$$|\psi(\vec{\gamma}, \vec{\beta})\rangle = U_M(\beta_p)U_C(\gamma_p) \dots U_M(\beta_1)U_C(\gamma_1) |s\rangle. \quad (15)$$

O loop de otimização clássica encarrega-se de ajustar de forma iterativa os parâmetros com o objetivo de minimizar o valor esperado do operador de custo. Demonstrou-se teoricamente que, no limite em que a profundidade $p \rightarrow \infty$, o algoritmo recupera a evolução adiabática exata, convergindo para a solução ótima global do problema (FARHI *et al.*, 2014; CROOKS, 2018).

Figura 6 – Exemplo de circuito para o QAOA (painel A) e o correspondente Grafo de Precedência Lógica dos Dados (painel B). O painel C exibe o grafo de interação para profundidade $p = 2$ e o grafo de conectividade do hardware. A otimização do escalonamento permite que as portas que compõem o operador $U_C(\gamma)$ sejam executadas em paralelo, reduzindo a profundidade efetiva do circuito.



Fonte: Adaptado de Guerreschi e Park (GUERRESCHI; PARK, 2018).

2.7 Relação com a Complexidade e Benchmarking Clássico

A proposição e o desenvolvimento de algoritmos variacionais como o VQE e o QAOA na era NISQ encontram como forte justificativa o gargalo ocasionado pela complexidade de tempo exponencial $\mathcal{O}(2^N)$ que caracteriza os métodos clássicos exatos ao lidarem com problemas NP-difíceis (GAREY; JOHNSON, 1979). A principal diretriz da computação quântica aplicável a esse cenário não reside na busca por soluções ótimas exatas instantâneas, mas sim na capacidade de fornecer soluções aproximadas de elevada qualidade com um custo computacional inferior ao de métodos clássicos em instâncias complexas (PRESKILL, 2018; CEREZO *et al.*, 2021).

No que concerne ao problema do *Maximum Cut*, o principal paradigma de comparação no domínio clássico estabelece-se através do algoritmo de Goemans-Williamson (GW) (GOEMANS; WILLIAMSON, 1995). Este método fundamentado em programação semidefinida tem uma garantia teórica robusta com um fator de aproximação de aproximadamente 0,8785, estabelecendo um limite que nenhum algoritmo clássico em tempo polinomial consegue superar (HÅSTAD, 2001; WILLIAMSON; SHMOYS, 2011). Por conseguinte, o processo de *benchmarking* comparativo entre abordagens quânticas e o algoritmo GW configura um campo de intensa

atividade científica (WANG *et al.*, 2018).

Embora simulações numéricas teóricas indiquem que algoritmos quânticos possuem potencial para suplantam o fator de aproximação clássico de 0.878 mediante a profundidade do circuito quântico ($p > 1$), a correspondente implementação física em dispositivos NISQ reais depara-se com barreiras práticas (FARHI *et al.*, 2017; YAN *et al.*, 2024). O aumento da profundidade implica circuitos maiores, provocando o acúmulo de erros e acelerando os efeitos da decoerência quântica, o que pode degradar a qualidade dos resultados obtidos antes mesmo que o otimizador clássico convirja (PRESKILL, 2018; ENDO *et al.*, 2021).

Adicionalmente, os algoritmos quânticos variacionais enfrentam desafios intrínsecos ao processo de otimização clássica (*barren plateaus*), caracterizado pelo desaparecimento exponencial dos gradientes da função de custo à medida que a escala do sistema cresce, dificultando severamente o treinamento dos parâmetros (MCCLEAN *et al.*, 2018; WURTZ; LOVE, 2021). Desse modo, o escopo da pesquisa atual direcionada ao MaxCut sob o paradigma quântico concentra-se fortemente no desenvolvimento de técnicas de redução de erros, simuladores avançados e arquiteturas de *ansatze* otimizadas para viabilizar vantagens práticas estáveis na era computacional.

2.8 Ferramentas Computacionais

A validação numérica e o desenvolvimento do pipeline experimental deste trabalho foi realizada em linguagem Python, utilizando um ecossistema de bibliotecas voltadas para modelagem combinatória, otimização convexa, análise estatística e computação quântica variacional. O ambiente de software foi estruturado para permitir o processamento híbrido, conectando a manipulação clássica de dados aos motores de simulação de circuitos quânticos.

Para a representação topológica e modelagem estrutural das instâncias do problema do MaxCut, utilizou-se a biblioteca NetworkX, responsável pela geração estocástica de grafos e extração de suas propriedades de conectividade. O suporte matemático e a manipulação de tensores assenta-se sobre a biblioteca NumPy, enquanto o gerenciamento, estruturação dos dados nas simulações de Monte Carlo foram consolidados com a biblioteca Pandas. Rotinas acessórias do núcleo nativo de Python, como os módulos `time` (cronometragem de alta precisão), `random` (geração pseudoaleatória), `itertools` (iteração combinatória exaustiva) e `warnings` (controle de alertas de execução), garantiram o fluxo de controle e o monitoramento de desempenho temporal dos algoritmos.

No domínio da otimização clássica e sintonia paramétrica, empregou-se a biblioteca CVXPY como interface algébrica para a Programação Semidefinida (SDP), permitindo a resolução do algoritmo de Goemans-Williamson. Complementarmente, a automação da busca hiperparamétrica para os níveis de repetição do circuito quântico foi delegada à biblioteca `scikit-optimize` (`skopt`), aplicando processos Gaussianos por meio do algoritmo de Otimização Bayesiana.

O núcleo quântico do ambiente foi desenvolvido inteiramente sob o ecossistema kit de desenvolvimento de software para computação quântica - Qiskit. O mapeamento do problema combinatório clássico para operadores quânticos foi introduzido pelo módulo `qiskit_optimization`, responsável por transpor a função de custo para o formalismo de um Hamiltoniano de Ising. A construção, parametrização e parametrização dos ansätze variacionais basearam-se nas classes fundamentais `QuantumCircuit`, `Parameter` e na biblioteca de circuitos estruturados `TwoLocal`.

Para a execução e convergência das funções de custo quânticas, utilizou-se o módulo `qiskit_algorithms`, com a estrutura do `SamplingVQE` integrada ao otimizador clássico COBYLA e ao gerenciador de reprodutibilidade estocástica `algorithm_globals`. Por fim, o processamento de alta fidelidade dos circuitos apoiou-se no pacote `qiskit_aer`, empregando o simulador de estado baseado em Matrizes de Produtos de Estados *Matrix Product State* - MPS através do `AerSimulator`, operado sob a arquitetura moderna de Primitivas V2 (`SamplerV2`) e precedido pela otimização lógica via gerenciador de transpilação `generate_preset_pass_manager`.

2.9 Métricas de Avaliação e Modelagem Estatística

A passagem da modelagem teórica para a análise computacional exige o estabelecimento de critérios para mensurar a qualidade das soluções e o custo operacional. Esta seção fundamenta os conceitos de aproximação, métodos estocásticos e técnicas preditivas que servem de base empírica e teórica para a avaliação de algoritmos híbridos e exatos.

2.9.1 Razão de Aproximação

Na resolução de problemas de otimização combinatória classificados como NP-difíceis, como o MaxCut, encontrar a solução ótima global torna-se tratável apenas para instâncias pequenas. Para avaliar a metodologia dos algoritmos quânticos variacionais, utiliza-se a métrica

de Razão de Aproximação (*Approximation Ratio* - AR) (VAZIRANI, 2013; WILLIAMSON; SHMOYS, 2011).

Para um problema de maximização, a razão de aproximação é definida formalmente como a razão entre o valor objetivo da solução obtida pelo algoritmo (C_{obtido}) e o valor da solução ótima teórica (C_{otimo}) (VAZIRANI, 2013):

$$AR = \frac{C_{obtido}}{C_{otimo}} \leq 1. \quad (16)$$

Quando a dimensão da topologia de rede dificulta o cálculo de C_{otimo} por métodos de força bruta, a teoria recomenda a adoção de um limite superior (*Upper Bound*). O relaxamento via Programação Semidefinida (SDP), conforme estabelecido por (GOEMANS; WILLIAMSON, 1995), fornece um limite superior estrito de 0,8785, garantindo uma métrica de AR segura, mesmo quando a solução exata é desconhecida.

2.9.2 Simulação Estocástica de Monte Carlo

A avaliação de sistemas computacionais que envolvem processos probabilísticos, seja na geração de topologias de rede via modelo de grafos de Erdős-Rényi (ERDOS; RENYI, 1959), seja na medição de estados em circuitos quânticos, exige o uso de ferramentas estatísticas robustas.

Para evitar que a análise de desempenho seja enviesada por flutuações amostrais atípicas, é padrão metodológico o emprego do método de simulação de Monte Carlo (MOTWANI; RAGHAVAN, 1996). Sua base estatística fundamenta-se na Lei dos Grandes Números, permitindo aproximar o valor esperado de uma variável aleatória por meio da média de múltiplas observações independentes. O estimador empírico é dado por (MOTWANI; RAGHAVAN, 1996):

$$\mathbb{E}[X] \approx \bar{X} = \frac{1}{k} \sum_{i=1}^k x_i, \quad (17)$$

onde k representa o número de amostras ou execuções independentes do sistema. Na prática, esse método assegura a estabilidade e a confiabilidade das métricas coletadas em ambientes ruidosos ou estocásticos (MOTWANI; RAGHAVAN, 1996).

2.9.3 Análise Assintótica e Expressividade Quântica

Para quantificar a eficiência dos algoritmos e classificar os problemas abordados na complexidade computacional, utiliza-se a análise assintótica. A ferramenta matemática padrão

para expressar o limite do tempo de execução de um algoritmo é a notação Big- $\mathcal{O}$ (BACHMANN, 1894; LANDAU, 1909; CORMEN *et al.*, 2009).

A função Big- $\mathcal{O}$ descreve o comportamento limitante de uma função quando a dimensão da entrada, denotada por n , tende ao infinito, ignorando constantes multiplicativas e termos de menor ordem. Formalmente, dadas duas funções $f(n)$ e $g(n)$, diz-se que (CORMEN *et al.*, 2009):

$$f(n) = \mathcal{O}(g(n)), \quad (18)$$

se, e somente se, existirem constantes positivas c e n_0 tais que (CORMEN *et al.*, 2009):

$$0 \leq f(n) \leq c \cdot g(n) \quad \text{para todo } n \geq n_0. \quad (19)$$

Essa definição matemática garante que a função $g(n)$ atue como um teto assintótico para $f(n)$. Na análise de algoritmos, $f(n)$ representa o número de operações ou o tempo de execução no pior caso. Algoritmos pertencentes à classe de complexidade P possuem tempo de execução limitado por um polinômio, expresso como $\mathcal{O}(n^k)$, onde k é uma constante (SIPSER, 2012; CORMEN *et al.*, 2009).

Conforme ilustrado na Figura 7, as curvas de crescimento polinomial e exponencial explicam a fronteira da tratabilidade algorítmica. Em contrapartida, na ausência de soluções eficientes, problemas NP-difíceis exigem abordagens de busca exaustiva. Nesses cenários, o espaço de soluções cresce de forma combinatória, resultando em algoritmos com complexidade de tempo exponencial, denotada por $\mathcal{O}(2^n)$ (GAREY; JOHNSON, 1979; KORTE; VYGEN, 2012).

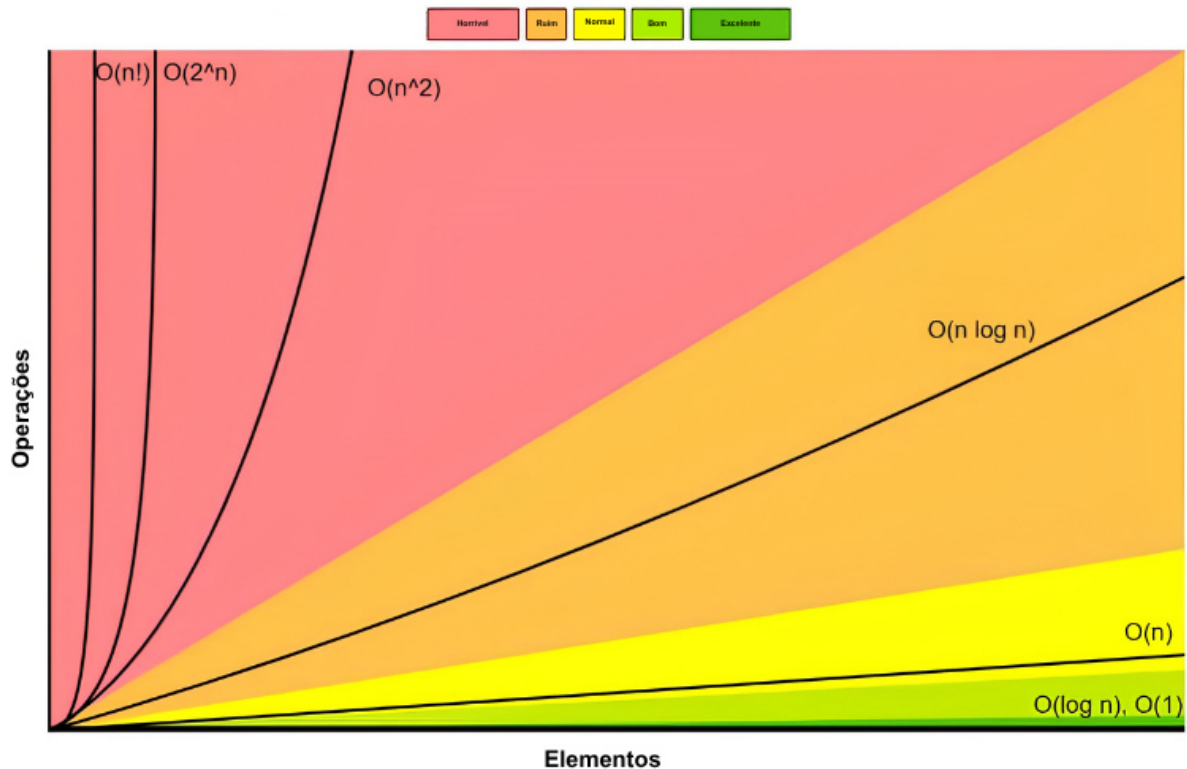
É exatamente o gargalo imposto pela barreira da complexidade $\mathcal{O}(2^n)$ que caracteriza a intratabilidade computacional para grafos grandes. Essa limitação justifica a necessidade rigorosa de métodos de *benchmarking* e motiva a busca por novos paradigmas. A exploração da informação quântica e a execução de algoritmos variacionais em simuladores e hardware surgem, portanto, como alternativas estratégicas para contornar a explosão combinatória e buscar vantagens computacionais sobre os métodos clássicos tradicionais (NIELSEN; CHUANG, 2010).

Embora a notação assintótica (Big- $\mathcal{O}$) seja suficiente para descrever a complexidade temporal clássica, a parametrização de circuitos quânticos dita o tamanho do espaço de busca que o otimizador contínuo precisará explorar (CEREZO *et al.*, 2021).

Em arquiteturas de algoritmos como o QAOA e o VQE, o aumento do número de camadas (profundidade do *ansatz*) amplia a expressividade do circuito, permitindo-lhe mapear estados mais complexos do espaço de Hilbert. Contudo, na literatura quântica, é amplamente

documentado que o excesso de dimensionalidade frequentemente leva ao fenômeno dos *Barren Plateaus* — onde os gradientes decaem exponencialmente, estagnando o aprendizado (MC-CLEAN *et al.*, 2018; LAROCCA *et al.*, 2025). Assim, a contagem e o controle paramétrico constituem critérios analíticos vitais para equilibrar a qualidade potencial da solução e a viabilidade da convergência.

Figura 7 – Hierarquia de crescimento das funções de complexidade Big- $\mathcal{O}$.



Fonte: Adaptado de FreeCodeCamp².

2.9.4 Projeção Empírica via Regressão Analítica

Devido ao crescimento exponencial da memória e do tempo exigidos para a simulação clássica de vetores de estado quântico, a execução estrita em computadores de alto desempenho esbarra em limites físicos de dimensionalidade. Para inferir o comportamento dos algoritmos em instâncias de maior porte sem a necessidade de execução direta, a teoria estatística emprega a modelagem preditiva por regressão analítica (GARETH *et al.*, 2013).

A regressão atua ajustando funções contínuas (como modelos polinomiais ou exponenciais) aos dados discretos de tempo e uso de recursos coletados empiricamente em cenários

² Disponível em: <<https://www.freecodecamp.org/news/big-o-cheat-sheet-time-complexity-chart/>>. Acesso em: 01 de maio de 2026.

menores. Seja $T(N)$ a variável de custo observada para um tamanho N , determinam-se os coeficientes $\vec{\beta}$ que minimizam a soma dos resíduos de um modelo $f(N, \vec{\beta})$ (GARETH *et al.*, 2013):

$$\min_{\vec{\beta}} \sum_i \left[T(N_i) - f(N_i, \vec{\beta}) \right]^2. \quad (20)$$

Uma vez validado o ajuste estatístico da curva, torna-se possível extrapolar a tendência para domínios superiores (GARETH *et al.*, 2013).

3 METODOLOGIA

Para avaliar as abordagens clássicas e quânticas na resolução do problema MaxCut, desenvolveu-se um ambiente experimental estruturado em linguagem Python. Esta seção detalha o fluxo de trabalho metodológico, desde a geração das instâncias de rede até a execução dos algoritmos e a projeção estatística dos resultados.

3.1 Caracterização do Método de Pesquisa

A pesquisa caracteriza-se como quantitativa e experimental, conduzida por meio de simulação computacional. O método baseia-se na execução comparativa de algoritmos exatos, heurísticos e quânticos variacionais sob as mesmas condições de estresse combinatorial. Os grafos atuam como modelos matemáticos de topologias de rede de telecomunicações, em que os vértices representam os nós de comunicação e as arestas denotam os enlaces físicos ou lógicos sujeitos a particionamento.

3.2 Modelagem das Instâncias de Rede

A geração das topologias foi implementada com o suporte da biblioteca *NetworkX* (HAGBERG *et al.*, 2008), utilizando o modelo de grafos aleatórios de Erdős-Rényi (ERDOS; RENYI, 1959).

Para padronizar a complexidade do problema e simular cenários de alta densidade de enlaces, a densidade de todos os grafos nesta etapa foi fixada em 60%. Dessa forma, o número total de arestas $|E|$ de uma instância com N vértices obedece estritamente à relação:

$$|E| = \left\lfloor 0,6 \times \frac{N(N-1)}{2} \right\rfloor. \quad (21)$$

Visando uma aproximação com redes de comunicação reais — onde os enlaces possuem larguras de banda ou latências distintas —, adotou-se um modelo de grafos ponderados. O peso de cada aresta foi definido aleatoriamente a partir de uma distribuição uniforme no intervalo contínuo $[0,5; 2,0]$, arredondado para duas casas decimais.

3.3 Escopo Experimental e Amostragem (Método de Monte Carlo)

Devido à natureza estocástica da geração de grafos e da simulação de medidas quânticas, o experimento foi projetado utilizando o método de simulação de Monte Carlo. O

domínio do tamanho do problema foi estabelecido no intervalo discreto de $4 \leq N \leq 20$ nós.

Para garantir validade estatística e atenuar flutuações, foram geradas 10 instâncias totalmente independentes para cada incremento de N . Todos os algoritmos solucionadores operaram concorrentemente sobre a mesma matriz de adjacência.

A reprodutibilidade exata do método de Monte Carlo foi assegurada por um sistema de sementes pseudoaleatórias (*seeds*). Partindo de uma semente-base $S_{base} = 10598$, a semente de cada iteração foi computada pela expressão $S_{atual} = S_{base} + (N \times 100) + A$, em que $A \in \{1, \dots, 10\}$ representa o valor da amostra. Esse número alimentou simultaneamente as rotinas do numpy, da biblioteca random e do simulador quântico Qiskit.

3.4 Configuração dos Algoritmos Solucionadores

O núcleo foi segmentado em quatro frentes de processamento, executadas em sequência para cada topologia de rede gerada:

1. **Força Bruta (Baseline Exato):** Solução computada pela enumeração iterativa de todo o espaço binário de partições. A complexidade $\mathcal{O}(2^{N-1})$ foi atingida fixando-se a polaridade do primeiro nó, extraindo o corte máximo global de forma determinística para validar os métodos até $N = 20$.
2. **Goemans-Williamson (GW):** Otimização por relaxamento de Programação Semidefinida (SDP) (GOEMANS; WILLIAMSON, 1995), implementada via biblioteca CVXPY (DIAMOND; BOYD, 2016) com o *solver Splitting Conic Solver* - SCS. O resultado contínuo foi submetido a um arredondamento de hiperplano aleatório em um *loop* de $\max(100, 10N)$ iterações, guardando o maior corte encontrado e o limite superior teórico (*upper bound*) da matriz SDP.
3. **Variational Quantum Eigensolver (VQE):** Executado via primitiva `SamplerV2` do Qiskit sob o modo *Matrix Product State* (MPS). O *ansatz* foi do tipo `TwoLocal` com profundidade estática de 3 repetições, totalizando $4N$ parâmetros clássicos, otimizados pelo algoritmo COBYLA com limite configurado em $\max(1000, 100N^2)$ iterações.
4. **Quantum Approximate Optimization Algorithm (QAOA):** Modelado sobre o ambiente *Qiskit* via `PassManager` otimizado para o `SamplerV2`. Para maximizar a relação custo-benefício, o número de camadas p foi otimizado antes da execução por meio de Otimização Bayesiana (Processos Gaussianos via `scikit-optimize`), prospectando o intervalo $p \in [1, 10]$ durante 15 chamadas.

3.5 Coleta de Métricas e Regressão Preditiva

Durante a execução do *loop* de Monte Carlo, os dados brutos de desempenho foram catalogados e exportados. As três variáveis primárias consistiram no tempo de processamento (em segundos), no número de parâmetros quânticos ajustados e na Razão de Aproximação (AR). O valor de cada instância foi calculado dividindo-se o corte obtido pelo corte referencial (Força Bruta para $N \leq 20$).

Como a validação limitou-se ao estresse combinatorial de $N = 20$, utilizou-se regressão para avaliar a escalabilidade dos algoritmos. A partir dos dados, modelos de regressão foram ajustados às curvas de tempo e dimensionalidade, permitindo projetar o custo e o comportamento das arquiteturas quânticas e clássicas até a marca de $N = 50$ vértices.

3.5.1 Infraestrutura Computacional

Para viabilizar a carga de processamento exigida pelas simulações quânticas, os testes foram submetidos em lote (*batch*) no Cluster Apollo, ambiente de computação de alto desempenho gerenciado pelo CENAPAD-UFC³.

O *headnode* (32 núcleos e 256 GB de RAM) foi utilizado para compilação via escalonador de tarefas SLURM. O processamento da simulação ocorreu nos *compute-nodes* exclusivos de CPU (Apollo01[1-6]), equipados com 256 núcleos e 512 GB de memória RAM. Com o objetivo de analisar o espaço sob recursos lógicos, os nós de aceleração gráfica (GPUs NVIDIA A100) foram isolados e não alocados para a simulação.

³ Centro Nacional de Processamento de Alto Desempenho da Universidade Federal do Ceará. Disponível em: <<https://cenapad.ufc.br/>>.

4 IMPLEMENTAÇÃO COMPUTACIONAL

Este capítulo descreve a engenharia de software empregada para os modelos teóricos em um ambiente de execução automatizado. A arquitetura do código foi projetada modularmente, garantindo o isolamento dos dados, dos solucionadores de otimização e do módulo estatístico.

4.1 Arquitetura do Código

A arquitetura do software foi desenvolvida visando garantir a segregação entre as fases de geração de dados, os motores de otimização (clássicos e quânticos) e o armazenamento dos resultados. O ciclo do *script* principal segue um *pipeline* sequencial: inicialização de ambiente com controle de sementes, criação da topologia, simulação para os quatro solucionadores (Força Bruta, GW, VQE e QAOA); por fim, efetua-se a coleta das métricas computacionais, como tempo de execução, valor do corte e número de iterações ou parâmetros envolvidos. O fluxograma do código completo pode ser visualizado na Figura 8.

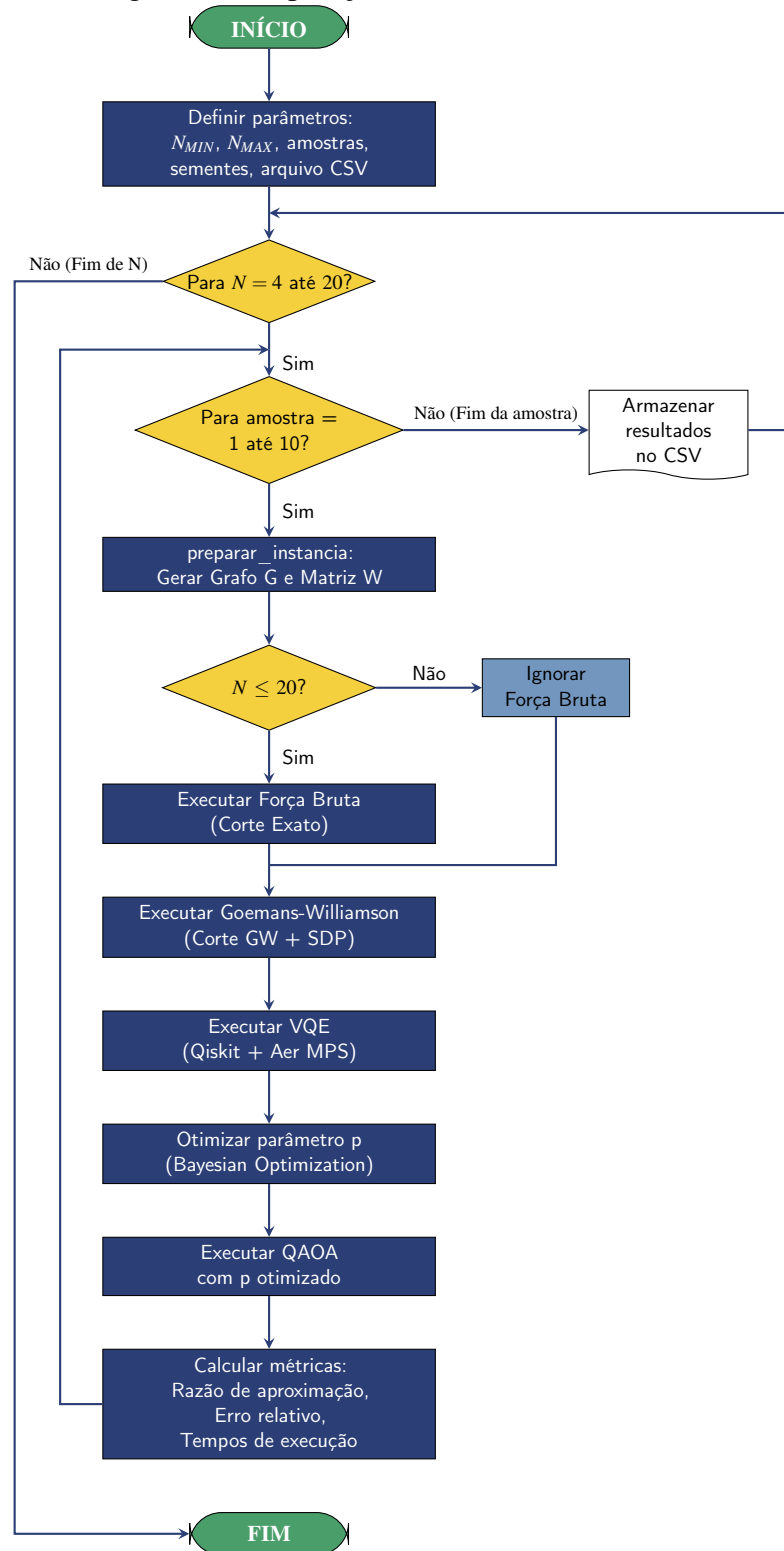
4.2 Pré-processamento dos Grafos

A etapa de pré-processamento é responsável por instanciar as topologias físicas. Por meio da biblioteca *NetworkX* (HAGBERG *et al.*, 2008), um grafo aleatório do tipo Erdős-Rényi é gerado com uma densidade calculada em 60%. Em seguida, o algoritmo passa por todas as arestas criadas e atribui a cada uma delas o peso de uma distribuição uniforme no intervalo contínuo $[0, 5; 2, 0]$. Essa estrutura de grafos ponderados é então convertida em uma matriz de adjacência bidimensional (matriz W), que atua como a estrutura de dados padronizada consumida por todos os algoritmos. O código-fonte encontra-se detalhado no Apêndice A.

4.3 Implementação da Força Bruta

Com o intuito de garantir um parâmetro de comparação seguro, a implementação foi programada para estabelecer o *baseline* ótimo global. Para atenuar a explosão combinatória clássica, a função de Força Bruta fixa estaticamente o estado binário do primeiro vértice do grafo, eliminando a redundância de partições espelhadas. Utilizando o módulo *itertools* nativo do Python, o algoritmo enumera 2^{N-1} permutações restantes. Para cada partição, a função de custo computa o peso do corte, registrando e retornando o valor máximo encontrado e o tempo de

Figura 8 – Fluxograma: Comparação de Métodos Max-Cut.



Fonte: Próprio autor.

execução cronometrado via `time.perf_counter()`. O código-fonte encontra-se detalhado no Apêndice B.

4.4 Implementação do Goemans-Williamson

Em contrapartida à busca exaustiva, a metodologia convexa de Goemans-Williamson foi modelada através do ecossistema CVXPY (DIAMOND; BOYD, 2016). A matriz de decisão bidimensional foi instanciada com a restrição de ser Semidefinida Positiva (*Positive Semidefinite* - PSD). A função objetivo de maximização do corte é submetida ao solucionador interno SCS (*Splitting Conic Solver*). Após encontrar o ótimo contínuo (que fornece o limite superior matemático), aplicou-se a técnica de arredondamento por hiperplano aleatório. O número de iterações de arredondamento foi parametrizado dinamicamente como $\max(100, 10 \times N)$ para assegurar a extração de uma partição discreta de alta qualidade. O código-fonte encontra-se detalhado no Apêndice C.

4.5 Implementação do VQE e QAOA

Os simuladores quânticos variacionais foram construídos utilizando o Qiskit (ALEKSANDROWICZ *et al.*, 2019). A matriz de adjacência é inicialmente traduzida para um Hamiltoniano de Ising via módulo `qiskit_optimization`.

Para o VQE, adotou-se o *ansatz* TwoLocal estruturado com portas de rotação *RY* e emaranhamento *CZ*, com profundidade fixa ($reps = 3$). A otimização dos parâmetros recaiu sobre o algoritmo clássico COBYLA, configurado com limite adaptativo de iterações de $\max(1000, 100 \times N^2)$.

Diferentemente do VQE, o circuito do QAOA foi desenhado de forma a permitir a flexibilidade paramétrica das camadas γ e β , conforme formalizado no estado quântico da Equação (15). O número de camadas (p) foi definido antes da execução do circuito mediante Otimização Bayesiana, gerenciada pela biblioteca `scikit-optimize`.

Ambos os algoritmos foram compilados sob um `PassManager` otimizado e submetidos à primitiva `SamplerV2`. Para reproduzir a restrição de recursos sem colapso de memória até $N = 20$, a simulação ocorreu no `AerSimulator` operando sob o método de representação de estados de produto de matriz (*Matrix Product State*-MPS). O código-fonte encontra-se detalhado no Apêndice D.

4.6 Organização dos Resultados

A gestão dos dados foi arquitetada de maneira tolerante a falhas. Empregou-se a biblioteca *pandas* para estruturar os dados gerados. A cada simulação isolada, foi armazenada uma lista de métricas com os valores de corte, tempos de processamento, ID da semente, limites superiores e dimensões. Ao concluir as 10 amostras dado o tamanho de rede N , o *DataFrame* em memória é exportado de forma incremental para um arquivo *.csv*. Esse comportamento previne a perda do conjunto de dados diante de potenciais interrupções no *cluster* de alto desempenho.

4.7 Geração de Gráficos e Tabelas

Os registros armazenados no arquivo CSV atuaram como fonte primária para o pós-processamento. Um *script* secundário, apoiado nas bibliotecas *matplotlib* e *seaborn*, foi empregado para calcular as estatísticas. O sistema gera três representações: a média da Razão de Aproximação (AR) traçada em relação aos limites ideais; a projeção da dimensionalidade paramétrica em escala logarítmica; e a regressão da complexidade temporal. Dessa forma, é possível projetar as tendências computacionais até a marca de $N = 50$ vértices.

4.8 Validação dos Experimentos

O rigor metodológico da simulação foi assegurado por meio de dois mecanismos de controle. Em um primeiro momento, a reprodutibilidade de todo o código foi garantida pela coordenação de sementes pseudoaleatórias calculadas por uma função que depende do identificador da amostra e do tamanho N . Isso assegura que qualquer execução futura gere matrizes idênticas e comportamentos de medição.

Somado a isso, conduziu-se a validação analítica da eficácia heurística e quântica contra margens de erro teóricas. Para instâncias até $N = 20$, os cortes encontrados foram validados contra a resposta extraída pela Força Bruta. Simultaneamente, o relaxamento Semidefinido (SDP) gerado pelo algoritmo de Goemans-Williamson foi coletado em todas as instâncias, atuando como certificador de limite superior (*Upper Bound*). Consequentemente, foi possível atestar com precisão a qualidade da Razão de Aproximação (AR) obtida pelos algoritmos quânticos.

5 RESULTADOS E DISCUSSÃO

A avaliação do desempenho dos algoritmos foi conduzida com base em três métricas principais: qualidade da solução, dimensionalidade paramétrica do modelo e projeção da escalabilidade temporal para instâncias de maior porte.

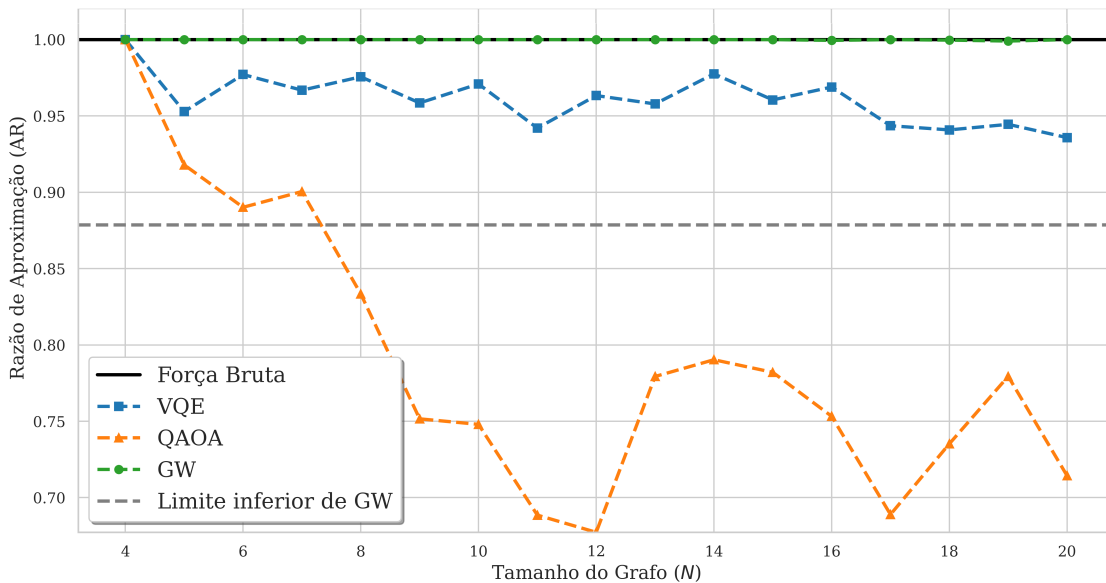
Os resultados foram obtidos para $4 \leq N \leq 20$. Para fins de visualização e análise de tendência, os valores apresentados para $N > 20$ nas figuras correspondem a uma interpolação baseada nos dados medidos, não representando novas simulações. Dessa forma, as projeções para tamanhos superiores do grafo devem ser interpretadas apenas como estimativas do comportamento esperado dos métodos avaliados.

Os resultados médios obtidos sobre 10 amostras independentes para cada valor de N são discutidos a seguir.

5.1 Qualidade da Solução e Razão de Aproximação

A razão de aproximação (*Approximation Ratio* – AR) mede a relação entre a pontuação do corte e o valor do corte ótimo global, obtido pelo método da força bruta. A Figura 9 apresenta o comportamento da AR para grafos com tamanhos variando de $N = 4$ a $N = 20$.

Figura 9 – Razão de aproximação média dos métodos avaliados em função do tamanho do grafo.



Fonte: Próprio autor.

Por definição, o método de força bruta mantém $AR = 1,0$ em todas as instâncias, ao

fornecer a solução exata. O algoritmo de Goemans-Williamson (GW) demonstrou estabilidade na resolução via Programação Semidefinida (SDP). Nas dimensões de rede correspondentes a $N = 15$, $N = 17$ e $N = 20$, o método atingiu a excelência em todas as amostras, registrando média de AR igual a 1,0000 e variância igual a 0,0000. Mesmo nos cenários de sutil dispersão provocada por flutuações, o algoritmo sustentou índices elevados (média de 0,9991 para $N = 19$).

Já nos algoritmos quânticos, observou-se uma diferença expressiva entre as abordagens. O VQE exibiu um ótimo desempenho: no limite inferior do intervalo ($N = 14$), alcançou uma média de AR de 0,9605. À medida que a dimensionalidade aumentou, manteve-se estável, atingindo 0,9358 sob uma variância de 0,0006 em $N = 20$, mantendo-se sempre muito acima do limite inferior do GW (0,8785).

Por sua vez, o QAOA apresentou um comportamento de maior oscilação e considerável degradação conforme o problema escalou. Embora tenha iniciado com $AR = 1,0000$ para $N = 4$, o algoritmo sofreu quedas, flutuando entre 0,7480 e 0,6773 para $N \geq 11$. Na maior parte dessas instâncias mais complexas, o QAOA performou abaixo do limite inferior do Goemans-Williamson. Esse declínio indica que, para o nível de profundidade (p) fixado neste estudo, o QAOA teve dificuldades em capturar a complexidade combinatória do espaço de soluções, necessitando de circuitos mais complexos para se equiparar à complexidade demonstrada pelo VQE.

5.2 Dimensionalidade Paramétrica

O volume de parâmetros necessário para cada modelo computacional influencia diretamente a viabilidade de alocação de memória e a complexidade. A Figura 10 mostra a evolução desse custo em escala logarítmica.

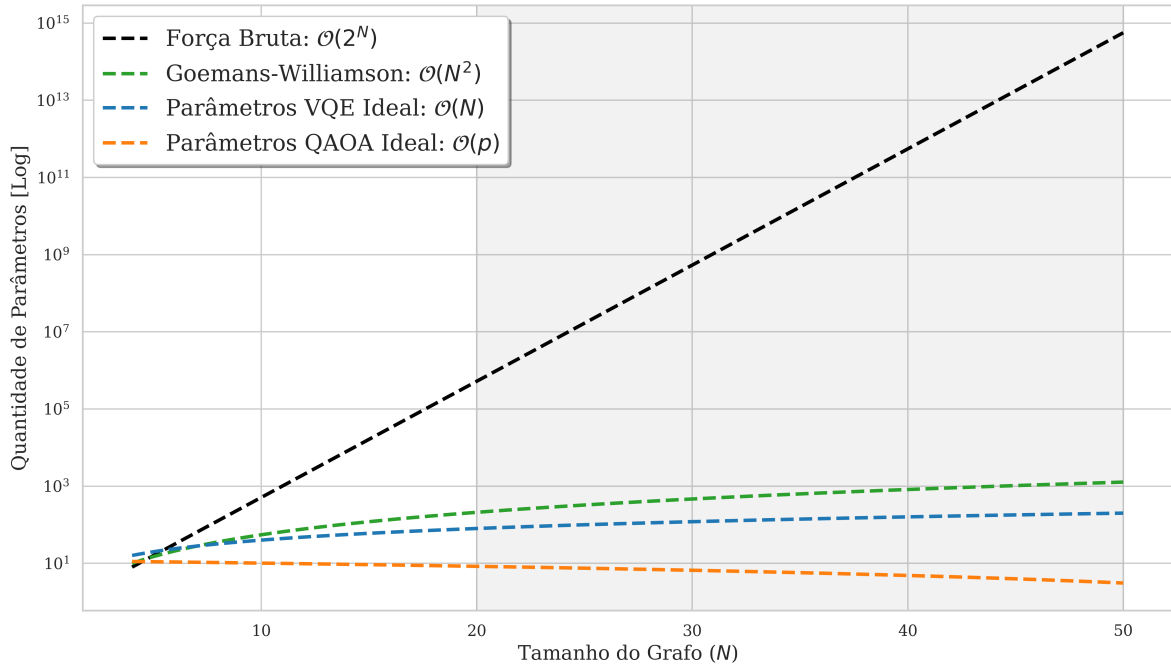
No caso da força bruta, o crescimento é exponencial com ordem $\mathcal{O}(2^N)$, o que a torna rapidamente inviável. O algoritmo de Goemans-Williamson apresenta um espaço de representação de ordem $\mathcal{O}(N^2)$, associado ao tamanho da matriz na relaxação semidefinida.

Nos modelos quânticos, a parametrização apresenta vantagens teóricas marcantes. O VQE exige um número de parâmetros que cresce linearmente, na ordem de $\mathcal{O}(N)$, compactando o espaço de busca e reduz o acúmulo de erros de portas lógicas em hardware.

Contudo, o grande destaque paramétrico reside no QAOA, que possui complexidade $\mathcal{O}(p)$, dependente do número de camadas do ansatz, e não da dimensão do grafo. Como mostrado nos resultados, a curva do QAOA mantém-se quase constante (na ordem de 10^1 parâmetros) para

qualquer valor de N . Embora essa parametrização reduza o espaço sobre o qual o otimizador clássico atua, ela também explica o menor desempenho na razão de aproximação (Figura 9), configurando um claro *trade-off* entre expressividade do modelo e volume paramétrico.

Figura 10 – Crescimento da dimensionalidade paramétrica dos métodos avaliados em escala logarítmica.



Fonte: Próprio autor.

5.3 Projeção de Escalabilidade Temporal

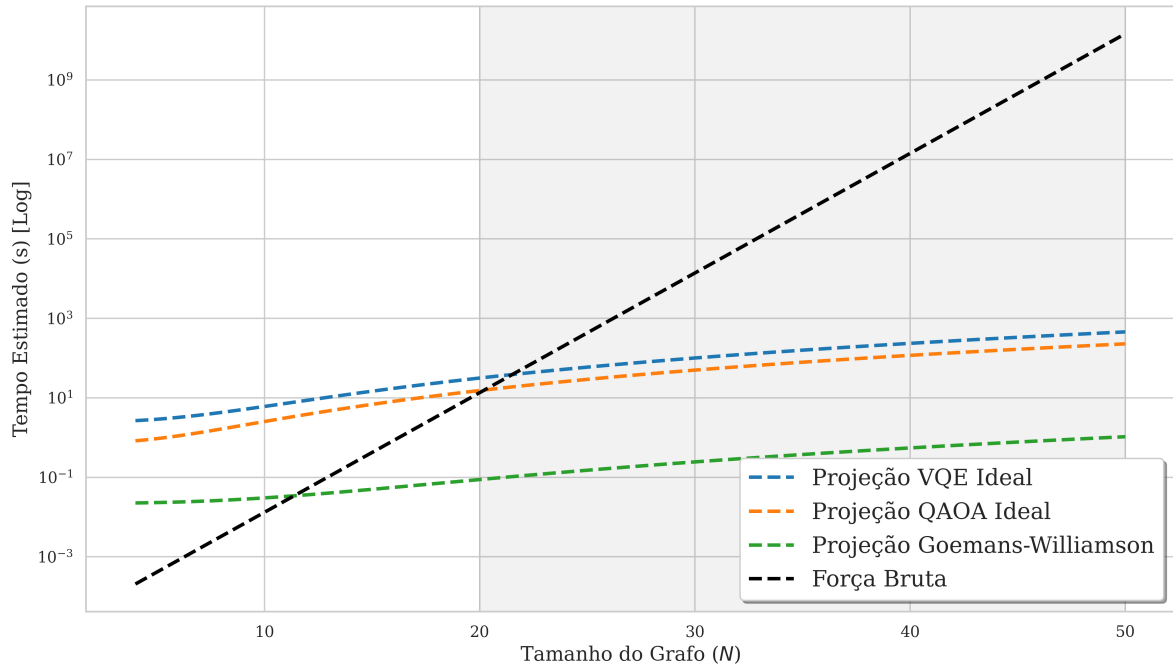
Para estimar o custo de execução em instâncias maiores e analisar os gargalos do ambiente clássico, a Figura 11 apresenta o tempo de execução para grafos até $N = 50$.

A força bruta exhibe a explosão combinatória esperada, com crescimento acentuado e valores impraticáveis. O método de Goemans-Williamson resolve todas as instâncias em tempos reduzidos, reafirmando sua eficiência polinomial.

A simulação clássica dos algoritmos quânticos evidencia um cenário desafiador. Apesar de suas parametrizações eficientes ($\mathcal{O}(N)$ e $\mathcal{O}(p)$), o custo computacional real engloba o rastreamento dos vetores de estado quântico no simulador clássico, o qual cresce exponencialmente com o número de qubits. Observa-se que o QAOA apresenta uma vantagem temporal em relação ao VQE, devido ao seu reduzido espaço de parâmetros ($p \ll N$), exigindo menos

avaliações pelo otimizador clássico a cada iteração. Entretanto, ambos os métodos quânticos ainda possuem tempos de execução significativos em simuladores clássicos para grandes grafos.

Figura 11 – Tempo de execução estimado dos métodos avaliados em função do tamanho do grafo (escala log).



Fonte: Próprio autor.

5.4 Síntese dos Resultados

Os resultados indicam que o algoritmo de Goemans-Williamson continua a oferecer o melhor equilíbrio prático entre qualidade da solução ($AR \approx 1,0$) e altíssima eficiência computacional.

Ao analisar as abordagens variacionais, evidenciou-se um *trade-off* entre a fidelidade da solução e a dimensionalidade paramétrica. O VQE demonstrou superioridade na taxa de aproximação, consolidando-se como um *ansatz* mais robusto para a convergência ao valor ótimo global, ainda que à custa de uma dimensão paramétrica superior ($\mathcal{O}(N)$). Em contrapartida, o QAOA destacou-se pela sua extrema compactação ($\mathcal{O}(p)$ parâmetros), resultando em tempos de execução menores no cenário. Contudo, essa eficiência parametrizada revelou-se insuficiente para capturar a complexidade combinatória da instância com a profundidade p limitada, resultando em quedas na qualidade do corte e desempenho frequentemente inferior à metodologia clássica

de Goemans-Williamson (GW).

A Tabela 1 resume o comportamento empírico no cenário de estresse combinatório. Fica evidente que, enquanto os algoritmos quânticos oferecem vantagens expressivas no mapeamento do problema (redução de variáveis), o custo da simulação os onera severamente em tempo.

Tabela 1 – Resultado da média e desvio-padrão ($\mu \pm \sigma$) para ($N = 14$ a 20).

N	Força Bruta		Goemans-Williamson		VQE		QAOA	
	Tempo [s]	Nº Parâm.	Tempo [s]	Nº Parâm.	Tempo [s]	Nº Parâm.	Tempo [s]	Nº Parâm.
20	$13,5224 \pm 0,0072$	524.288	$0,0873 \pm 0,0008$	210	$32,4780 \pm 18,2541$	80	$20,7356 \pm 194,5994$	12
19	$6,0959 \pm 0,0030$	262.144	$0,0792 \pm 0,0008$	190	$26,7409 \pm 18,2541$	76	$13,4380 \pm 27,9528$	11
18	$2,7708 \pm 0,0002$	131.072	$0,0634 \pm 0,0000$	171	$20,7592 \pm 7,4287$	72	$6,6111 \pm 41,6602$	7
17	$1,2582 \pm 0,0000$	65.536	$0,0659 \pm 0,0008$	153	$19,6308 \pm 6,7929$	68	$7,8189 \pm 18,7566$	9
16	$0,5752 \pm 0,0001$	32.768	$0,0517 \pm 0,0000$	136	$17,3843 \pm 3,6169$	64	$6,2414 \pm 16,9912$	8
15	$0,2706 \pm 0,0001$	16.384	$0,0471 \pm 0,0000$	120	$13,8878 \pm 3,4166$	60	$5,2408 \pm 16,1303$	8
14	$0,1136 \pm 0,0000$	8.192	$0,0495 \pm 0,0009$	105	$16,6528 \pm 5,3875$	56	$8,3050 \pm 37,0225$	9

Nota: O número de parâmetros do QAOA depende das p -camadas definidas no experimento. Valores exibidos representam a média observada. O tempo exato acompanhou proporções inferiores às do VQE (ver projeções na Figura 11).

Dessa forma, os resultados sugerem que a principal limitação atual das abordagens variacionais quânticas não reside apenas na formulação algorítmica, mas também na execução em simuladores. Em ambientes NISQ (*Noisy Intermediate-Scale Quantum*), a obtenção de vantagem prática dependerá de hardware quântico real, aliada a estratégias para aprofundamento do circuito do QAOA ou técnicas mais avançadas de otimização de parâmetros (CEREZO *et al.*, 2021; KANDALA *et al.*, 2017).

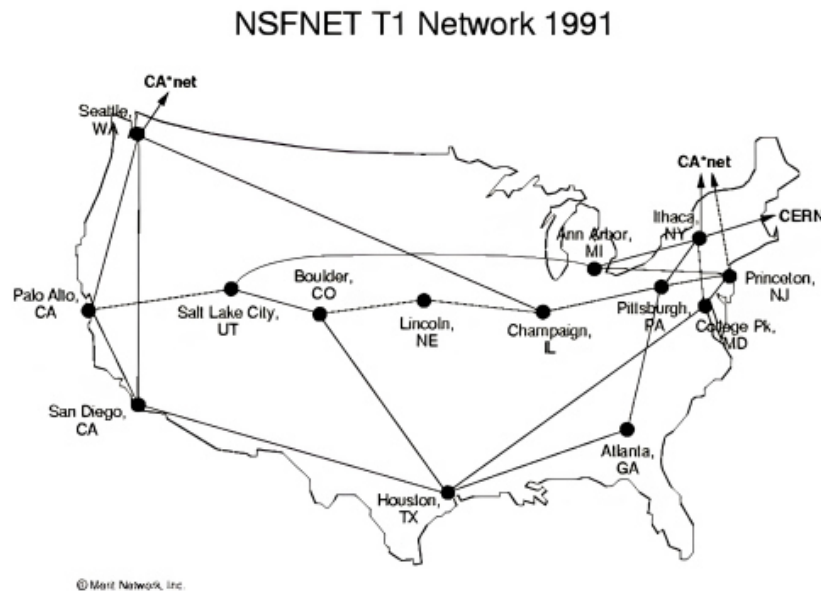
5.5 Estudo de Caso: Topologia NSFNET (1991)

Para consolidar a análise entre os métodos clássicos e quânticos em um cenário real de redes de telecomunicações, o problema do Corte Máximo (*MaxCut*) foi aplicado à topologia de *backbone* da *National Science Foundation Network* (NSFNET) de 1991. Trata-se de uma infraestrutura histórica de longa distância (WAN) composta por 14 nós e 20 arestas, conforme ilustrado no diagrama de referência original da Figura 12.

Para a aplicação dos algoritmos, a topologia foi mapeada utilizando a biblioteca *NetworkX*. O algoritmo desenvolvido fixa as coordenadas bidimensionais de cada um dos 14 vértices para preservar a disposição geográfica da rede e define rigorosamente as 20 conexões físicas. Visando uma aproximação com redes de comunicação reais — onde os enlaces possuem larguras de banda ou latências distintas —, adotou-se um modelo de grafos ponderados. O peso de cada aresta foi definido aleatoriamente a partir de uma distribuição uniforme no intervalo

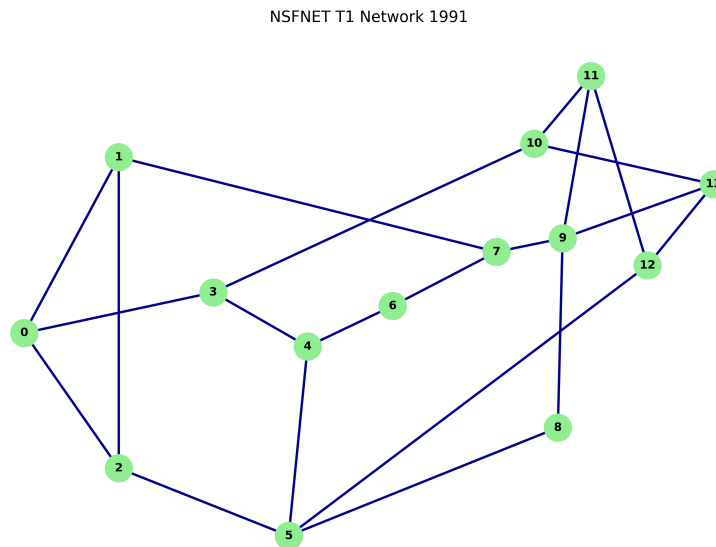
contínuo $[0,5;2,0]$, arredondado para duas casas decimais. A reconstrução final do grafo gerada por esse processo é apresentada na Figura 13.

Figura 12 – Diagrama original da topologia de *backbone* da NSFNET (1991).



Fonte: Adaptado de
<https://commons.wikimedia.org/wiki/File:NSFNET-backbone-T1.png>

Figura 13 – Reconstrução e modelagem em grafo da topologia NSFNET.



Fonte: Próprio autor.

A introdução desses pesos altera a dinâmica de otimização. Além disso, é necessário destacar o contraste estrutural deste estudo de caso em relação à metodologia de simulação anterior. Para padronizar a complexidade e simular cenários de alta conectividade nas rodadas de

grafos aleatórios representados pela equação 21, a densidade topológica havia sido fixada em 60%.

Em contrapartida, para a topologia empírica da NSFNET, calcula-se a densidade real do grafo (D), Equação (1), dada pela razão entre o número de arestas existentes ($|E| = 20$) e o limite máximo de conexões teóricas para $|V| = 14$ nós:

$$D = \frac{2|E|}{|V|(|V| - 1)} = \frac{2 \times 20}{14 \times 13} = \frac{40}{182} \approx 0,2198. \quad (22)$$

A densidade de aproximadamente 22% caracteriza a NSFNET como um grafo esparso. Em estruturas esparsas onde os pesos das arestas variam continuamente, a paisagem de energia do Hamiltoniano de Ising associado torna-se irregular, refletindo-se nos mínimos locais. Contudo, devido ao tamanho moderado do problema ($N = 14$), o algoritmo clássico de Goemans-Williamson (GW) via Relaxamento Semidefinido (SDP) consegue contornar essas flutuações com eficiência, alcançando consistentemente o valor exato do corte determinado pela Força Bruta. A Tabela 2 sintetiza o desempenho dos algoritmos, destacando as métricas críticas de Tempo de Execução e Razão de Aproximação (AR) obtidas a partir do ambiente de simulação.

Tabela 2 – Resultado da média e desvio-padrão ($\mu \pm \sigma$) para o estudo de caso NSF-NET.

Topologia	Força Bruta	Goemans-Williamson		VQE		QAOA	
	Tempo [s]	Tempo [s]	AR	Tempo [s]	AR	Tempo [s]	AR
NSFNET (1991)	$0,1487 \pm 0,0269$	$0,0916 \pm 0,0728$	$1,0000 \pm 0,0000$	$44,7732 \pm 9,7238$	$0,9388 \pm 0,0695$	$5,7592 \pm 1,4731$	$0,7447 \pm 0,2291$

Os resultados expostos na Tabela 2 revelam o impacto direto da esparsidade e da variação de pesos nos solucionadores quânticos variacionais. O VQE exibiu o maior tempo computacional médio ($\approx 44,77$ s), consequência do elevado número de avaliações de funções pelo otimizador clássico COBYLA para ajustar os parâmetros do circuito em uma paisagem de custo complexa, embora tenha garantido uma AR sólida de 0,9388.

Em contrapartida, o QAOA mostrou-se mais célere ($\approx 5,75$ s), mas sofreu uma penalidade severa na qualidade da solução, registrando uma AR média de 0,7447 com um desvio-padrão acentuado ($\pm 0,2291$). Essa oscilação e degradação no desempenho do QAOA ocorrem porque a distribuição não-uniforme de pesos dificulta a sincronização dos operadores, misturando-se ao longo das camadas do circuito, fazendo com que o algoritmo colapse em cortes submáximos na presença de topologias esparsas.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho investigou a aplicação e o desempenho de abordagens clássicas e quânticas na resolução do problema de particionamento de grafos (MaxCut), modelado sobre instâncias inspiradas em topologias de redes de telecomunicações. A análise metodológica baseou-se na avaliação da qualidade, da dimensionalidade e da escalabilidade temporal, comparando o método exato de Força Bruta, o algoritmo de Goemans-Williamson e as abordagens variacionais quânticas VQE e QAOA.

Os resultados demonstraram que, das abordagens clássicas, o algoritmo de Goemans-Williamson consolidou-se como o método com a melhor relação custo-benefício. Apoiado no relaxamento por Programação Semidefinida (SDP), o GW sustentou uma alta eficiência temporal e registrou uma Razão de Aproximação (AR) próxima de 1,0000 em quase todas as instâncias, confirmando a sua robustez para o particionamento de redes, tanto em cenários paramétricos de alta densidade como em topologias esparsas e ponderadas (evidenciado no estudo de caso da rede NSFNET). Por sua vez, o método de Força Bruta comportou-se perfeitamente como linha de base (*baseline*) exata; contudo, a sua complexidade exponencial inviabiliza-o para instâncias de maior magnitude.

No que concerne aos algoritmos quânticos, o estudo evidenciou o *trade-off* existente entre a fidelidade da solução e a compactação paramétrica:

- **VQE:** Consagrou-se como o *ansatz* mais robusto para convergência. Exigindo um volume paramétrico de crescimento linear na ordem de $\mathcal{O}(N)$, o modelo manteve estabilidade e uma Razão de Aproximação superior e constante, mesmo em experimentos mais complexos ($N = 20$).
- **QAOA:** Destacou-se por uma redução drástica e favorável na parametrização, operando na ordem de $\mathcal{O}(p)$, independentemente da dimensão do grafo. Contudo, essa compactação revelou-se insuficiente para capturar a densidade combinatória dos grafos à medida que estes aumentavam. Para a profundidade testada, o QAOA sofreu uma degradação na qualidade do corte — penalidade que se agravou perante a distribuição não uniforme de pesos nas arestas da topologia —, operando frequentemente abaixo do limite inferior estabelecido pelo algoritmo de GW.

Por fim, a principal limitação das abordagens quânticas atuais não reside nas suas formulações matemáticas, mas sim no gargalo imposto pela simulação clássica. O custo computacional necessário para rastrear os vetores de estado em CPUs convencionais exigiu tempos

de processamento excessivos para o VQE e o QAOA, anulando temporalmente as vantagens da redução paramétrica. Dessa forma, conclui-se que a real eficiência desses algoritmos variacionais permanece vinculada à execução nativa em hardwares quânticos físicos.

A partir dos dados compilados e das limitações computacionais observadas neste estudo, sugerem-se as seguintes vertentes para pesquisas futuras:

- **Implementação na Era NISQ:** Transpor os experimentos dos simuladores clássicos para QPUs físicas. Esta evolução eliminará a sobrecarga do rastreamento de vetores de estado, permitindo aferir o tempo real de convergência e avaliar a resiliência dos modelos perante os ruídos intrínsecos ao hardware.
- **Aprofundamento do *Ansatz* QAOA:** Investigar o comportamento do QAOA utilizando camadas mais profundas ($p > 1$) e novos otimizadores. O objetivo primordial é mitigar a degradação da Razão de Aproximação mapeada neste trabalho, verificando se circuitos de maior profundidade conseguem equiparar a precisão analítica do QAOA à do VQE sem comprometer a vantagem no tempo de execução.
- **Aplicação de Técnicas de Mitigação de Erros:** No contexto da transição para os processadores NISQ, será imperativo integrar técnicas formais de mitigação de erros quânticos aos circuitos. Projetar e avaliar de que forma estas ferramentas lidam com as topologias altamente complexas será fundamental para viabilizar implementações fidedignas.
- **Exploração de Casos de Uso Dinâmicos em Telecomunicações:** Expandir a modelagem de base (MaxCut) muito além das topologias estáticas. Os algoritmos poderão ser customizados para resolver problemas infraestruturais dinâmicos e complexos, tais como o roteamento de tráfego adaptativo, a alocação dinâmica de espectro em redes ópticas e o *network slicing*, visando suprir as rigorosas exigências do 5G e das futuras arquiteturas 6G.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALEKSANDROWICZ, G.; ALEXANDER, T.; BARKOUTSOS, P.; BELLO, L.; BEN-HAIM, Y.; BUCHER, D.; CABRERA-HERNÁNDEZ, F. J.; CARBALLO-FRANQUIS, J.; CHEN, A.; CHEN, C.-F. *et al.* **Qiskit: An open-source framework for quantum computing**. 2019.
- ARORA, S.; BARAK, B. **Computational complexity: a modern approach**. [S.l.]: Cambridge University Press, 2009.
- BACHMANN, P. **Die analytische zahlentheorie**. [S.l.]: Teubner, 1894.
- BARAHONA, F.; GRÖTSCHEL, M.; JÜNGER, M.; REINELT, G. An application of combinatorial optimization to statistical physics and circuit layout design. **Operations Research, INFORMS**, v. 36, n. 3, p. 493–513, 1988.
- BOYD, S.; VANDENBERGHE, L. **Convex optimization**. [S.l.]: Cambridge university press, 2004.
- CEREZO, M.; ARRASMITH, A.; BABBUSH, R.; BENJAMIN, S. C.; ENDO, S.; FUJII, K.; MCCLEAN, J. R.; MITARAI, K.; YUAN, X.; CINCIO, L. *et al.* Variational quantum algorithms. **Nature Reviews Physics**, Nature Publishing Group UK London, v. 3, n. 9, p. 625–644, 2021.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to Algorithms**. 3. ed. Cambridge, MA: MIT Press, 2009.
- CROOKS, G. E. Performance of the quantum approximate optimization algorithm on the maximum cut problem. **arXiv preprint arXiv:1811.08419**, 2018.
- DEZA, M. M.; LAURENT, M. **Geometry of Cuts and Metrics**. Berlin, Heidelberg: Springer-Verlag, 1997.
- DIAMOND, S.; BOYD, S. Cvxpy: A python-embedded modeling language for convex optimization. **Journal of Machine Learning Research**, v. 17, n. 83, p. 1–5, 2016.
- DIESTEL, R. **Graph Theory**. 5. ed. Heidelberg: Springer-Verlag, 2017. (Graduate Texts in Mathematics, v. 173).
- ENDO, S.; CAI, Z.; BENJAMIN, S. C.; YUAN, X. Hybrid quantum-classical algorithms and quantum error mitigation. **Journal of the Physical Society of Japan**, v. 90, n. 3, p. 032001, 2021.
- ERDOS, P.; RENYI, A. On random graphs. *publicationes mathematicae (debrecen)*, vol. 6. 1959.
- FARHI, E.; GOLDSTONE, J.; GUTMANN, S. A quantum approximate optimization algorithm. **arXiv preprint arXiv:1411.4028**, 2014.
- FARHI, E.; GOLDSTONE, J.; GUTMANN, S.; NEVEN, H. Quantum algorithms for fixed qubit architectures. **arXiv preprint arXiv:1703.06199**, 2017.
- GARETH, J.; DANIELA, W.; TREVOR, H.; ROBERT, T. **An introduction to statistical learning: with applications in R**. [S.l.]: Springer, 2013.
- GAREY, M.; JOHNSON, D. **Computers and Intractability: A Guide to the Theory of NP-completeness**. W. H. Freeman, 1979. (Mathematical Sciences Series). ISBN 9780716710448. Available at: <<https://books.google.com.br/books?id=fjxGAQAIAAJ>>.

GOEMANS, M. X.; WILLIAMSON, D. P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. **J. ACM**, Association for Computing Machinery, New York, NY, USA, v. 42, n. 6, p. 1115–1145, Nov. 1995. ISSN 0004-5411. Available at: <<https://doi.org/10.1145/227683.227684>>.

GUERRESCHI, G. G.; PARK, J. Two-step approach to scheduling quantum circuits. **Quantum Sci. Technol.**, v. 3, n. 4, p. 045003, 2018.

HAGBERG, A. A.; SCHULT, D. A.; SWART, P. J.; HAGBERG, J. Exploring network structure, dynamics, and function using networkx. **Proceedings of the Python in Science Conference**, 2008. Available at: <<https://api.semanticscholar.org/CorpusID:16050699>>.

HÅSTAD, J. Some optimal inapproximability results. **Journal of the ACM (JACM)**, ACM New York, NY, USA, v. 48, n. 4, p. 798–859, 2001.

KANDALA, A.; MEZZACAPO, A.; TEMME, K.; TAKITA, M.; BRINK, M.; CHOW, J. M.; GAMBETTA, J. M. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. **Nature**, Nature Publishing Group, v. 549, n. 7671, p. 242–246, 2017.

KARP, R. M. Reducibility among combinatorial problems. In: _____. **Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York**. Boston, MA: Springer US, 1972. p. 85–103. ISBN 978-1-4684-2001-2. Available at: <https://doi.org/10.1007/978-1-4684-2001-2_9>.

KORTE, B.; VYGEN, J. **Combinatorial Optimization: Theory and Algorithms**. 5. ed. Berlin: Springer, 2012.

LANDAU, E. **Handbuch der Lehre von der Verteilung der Primzahlen**. Leipzig: B.G. Teubner, 1909.

LAROCCA, M.; THANASILP, S.; WANG, S.; SHARMA, K.; BIAMONTE, J.; COLES, P. J.; CINCIO, L.; MCCLEAN, J. R.; HOLMES, Z.; CEREZO, M. Barren plateaus in variational quantum computing. **Nature Reviews Physics**, Nature Publishing Group UK London, v. 7, n. 4, p. 174–189, 2025.

LUCAS, A. Ising formulations of many np problems. **Frontiers in Physics**, Frontiers, v. 2, p. 74887, 2014.

LUENBERGER, D. G.; YE, Y. *et al.* **Linear and nonlinear programming**. [S.l.]: Springer, 1984.

MCCLEAN, J. R.; BOIXO, S.; SMELYANSKIY, V. N.; BABBUSH, R.; NEVEN, H. Barren plateaus in quantum neural network training landscapes. **Nature communications**, Nature Publishing Group UK London, v. 9, n. 1, p. 4812, 2018.

MCCLEAN, J. R.; ROMERO, J.; BABBUSH, R.; ASPURU-GUZI, A. The theory of variational hybrid quantum-classical algorithms. **New Journal of Physics**, IOP Publishing, v. 18, n. 2, p. 023023, 2016.

MCCLEAN, J. R.; ROMERO, J.; BABBUSH, R.; ASPURU-GUZI, A. The theory of variational hybrid quantum-classical algorithms. **New Journal of Physics**, IOP Publishing, v. 18, n. 2, p. 023023, 2016.

MOTWANI, R.; RAGHAVAN, P. **Randomized algorithms**. [S.l.]: ACM New York, NY, USA, 1996. 33–37 p.

NESTEROV, Y.; NEMIROVSKII, A. **Interior-point polynomial algorithms in convex programming**. [S.l.]: SIAM, 1994.

NIELSEN, M. A.; CHUANG, I. L. **Quantum computation and quantum information**. [S.l.]: Cambridge university press, 2010.

PAPADIMITRIOU, C. H. Computational complexity. In: **Encyclopedia of computer science**. [S.l.: s.n.], 2003. p. 260–265.

PERUZZO, A.; MCCLEAN, J.; SHADBOLT, P.; YUNG, M.-H.; ZHOU, X.-Q.; LOVE, P. J.; ASPURU-GUZI, A.; O'BRIEN, J. L. A variational eigenvalue solver on a photonic quantum processor. **Nature communications**, Nature Publishing Group UK London, v. 5, n. 1, p. 4213, 2014.

PRESKILL, J. Quantum computing in the nisq era and beyond. **Quantum**, Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, v. 2, p. 79, 2018.

RESENDE, M.; PARDALOS, P. **Handbook of Optimization in Telecommunications**. Springer US, 2006. ISBN 9780387306629. Available at: <<https://books.google.com.br/books?id=6aERjwEACAAJ>>.

SIPSER, M. **Introduction to the theory of computation**. Cengage Learning. [S.l.: s.n.], 2012.

STĘCHŁY, M.; BASHIGE, N.; CHOJECKI, P. Approaching graph problems with continuous variable quantum computing. **arXiv preprint arXiv:1906.07047**, 2019.

TOSCANI, L. V.; VELOSO, P. A. Complexidade de algoritmos. **Sagra-Luzzatto. Porto Alegre**, 2001.

VANDENBERGHE, L.; BOYD, S. Semidefinite programming. **SIAM Review**, v. 38, n. 1, p. 49–95, 1996. Available at: <<https://doi.org/10.1137/1038003>>.

VAZIRANI, V. **Approximation Algorithms**. Springer Berlin Heidelberg, 2002. ISBN 9783540653677. Available at: <<https://books.google.co.uk/books?id=EILqAmzKgYIC>>.

VAZIRANI, V. **Approximation Algorithms**. Springer Berlin Heidelberg, 2013. (Computer Science). ISBN 9783662045657. Available at: <<https://books.google.com.br/books?id=bJmqCAAAQBAJ>>.

WANG, Z.; HADFIELD, S.; JIANG, Z.; RIEFFEL, E. G. Quantum approximate optimization algorithm for maxcut: A fermionic view. **Phys. Rev. A**, American Physical Society, v. 97, p. 022304, Feb 2018.

WECKER, D.; HASTINGS, M. B.; TROYER, M. Training a quantum optimizer. **Physical Review A**, APS, v. 94, n. 2, p. 022309, 2016.

WILLIAMSON, D.; SHMOYS, D. **The Design of Approximation Algorithms**. Cambridge University Press, 2011. ISBN 9781107214514. Available at: <https://books.google.com.br/books?id=T2_XzgEACAAJ>.

WOLKOWICZ, H.; SAIGAL, R.; VANDENBERGHE, L. **Handbook of semidefinite programming: theory, algorithms, and applications**. [S.l.]: Springer Science & Business Media, 2012.

WURTZ, J.; LOVE, P. Maxcut quantum approximate optimization algorithm performance guarantees for $p > 1$. **Phys. Rev. A**, American Physical Society, v. 103, p. 042612, Apr 2021. Available at: <<https://link.aps.org/doi/10.1103/PhysRevA.103.042612>>.

YAN, G.; WU, W.; CHEN, Y.; PAN, K.; LU, X.; ZHOU, Z.; WANG, Y.; WANG, R.; YAN, J. Quantum Circuit Synthesis and Compilation Optimization: Overview and Prospects. 6 2024.

APÊNDICE A – GERAÇÃO DE INSTÂNCIAS

Este apêndice apresenta as funções base responsáveis pela geração dos grafos aleatórios e otimização de hiperparâmetros.

Código-fonte 1 – função de instanciação e otimização

```

1
2 def otimizar_hiperparametros_qaoa(matriz_w, n_nos):
3     space = [Integer(1, 10, name='p_reps')]
4
5     @use_named_args(space)
6     def objetivo_qaoa(p_reps):
7         corte, _, _ = rodar_qaoa_qiskit(matriz_w, n_nos,
8             p_reps=int(p_reps))
9         return -corte
10
11     res = gp_minimize(
12         objetivo_qaoa, space, n_calls=15, random_state=42,
13         base_estimator="GP"
14     )
15     return int(res.x[0])
16
17 def preparar_instancia(n_nos, densidade=0.6, aleatorio=True):
18     max_arestas = n_nos * (n_nos - 1) // 2
19     m_arestas = int(max_arestas * densidade)
20     G = nx.gnm_random_graph(n_nos, max(1, m_arestas))
21
22     # Atribuicao de pesos aleatorios as arestas
23     for (u, v) in G.edges():
24         G.edges[u, v]['weight'] = round(random.uniform(0.5,
25             2.0), 2) if aleatorio else 1.0

```

```
24     # Construcao da Matriz de Adjacencia
25     matriz_w = np.zeros([n_nos, n_nos])
26     for i in range(n_nos):
27         for j in range(n_nos):
28             temp = G.get_edge_data(i, j, default=0)
29             if temp != 0:
30                 matriz_w[i, j] = temp["weight"]
31
32     return G, matriz_w
```

APÊNDICE B – IMPLEMENTAÇÃO DA FORÇA BRUTA

Implementação exata para extração do *baseline* global. O espaço de busca combinatorial é varrido fixando-se a polaridade do primeiro nó.

Código-fonte 2 – Rotina exata determinística do problema MaxCut

```
1 def maxcut_forca_bruta(matriz_w, n_nos):  
2     t0 = time.perf_counter()  
3     melhor_corte = 0  
4  
5     for particao_tupla in itertools.product([0, 1], repeat=  
6         n_nos - 1):  
7         particao = (0,) + particao_tupla  
8         corte_atual = sum(matriz_w[i][j] for i in range(  
9             n_nos) for j in range(i+1, n_nos) if particao[i]  
10             != particao[j])  
11  
12         if corte_atual > melhor_corte:  
13             melhor_corte = corte_atual  
14  
15     return melhor_corte, time.perf_counter() - t0
```

APÊNDICE C – IMPLEMENTAÇÃO DO GOEMANS-WILLIAMSON

Heurística clássica estruturada sobre relaxamento Semidefinido Positivo (SDP) através do *solver* SCS da biblioteca CVXPY, seguida do método de arredondamento em hiperplano aleatório.

Código-fonte 3 – Relaxamento convexo e arredondamento estocástico do algoritmo

GW

```

1 def maxcut_goemans_williamson(matriz_w, n_nos,
2     multiplicador_arredondamento=10):
3     t0 = time.perf_counter()
4     X = cp.Variable((n_nos, n_nos), PSD=True)
5     objetivo = sum(matriz_w[i][j] * 0.5 * (1 - X[i, j]) for
6         i in range(n_nos) for j in range(i+1, n_nos) if
7         matriz_w[i][j] > 0)
8     restricoes = [X[i, i] == 1 for i in range(n_nos)]
9     prob = cp.Problem(cp.Maximize(objetivo), restricoes)
10    prob.solve(solver=cp.SCS, verbose=False)
11
12    X_otimo = (X.value + X.value.T) / 2
13    eigvals, eigvecs = np.linalg.eigh(X_otimo)
14    V = eigvecs @ np.diag(np.sqrt(np.maximum(eigvals, 0)))
15
16    melhor_corte = 0
17    limite_arredondamentos = max(100, n_nos *
18        multiplicador_arredondamento)
19
20    for _ in range(limite_arredondamentos):
21        r = np.random.randn(n_nos)
22        particao = [1 if np.dot(V[i], r) > 0 else 0 for i
23            in range(n_nos)]
24        corte_atual = sum([matriz_w[i][j] for i in range(

```

```
                n_nos) for j in range(i+1, n_nos) if particao[i]
                    != particao[j]])
21         melhor_corte = max(melhor_corte, corte_atual)
22
23     return melhor_corte, prob.value, time.perf_counter() -
        t0
```

APÊNDICE D – IMPLEMENTAÇÃO DO VQE E QAOA

A codificação dos simuladores quânticos parametrizados construída sobre o ecossistema *Qiskit*. Destaca-se a primitiva `SamplerV2` emulando o modo MPS e a otimização de hiperparâmetros (Otimização Bayesiana) para estruturação das camadas do circuito QAOA.

Código-fonte 4 – Motores quânticos variacionais adaptados para a primitiva `SamplerV2` e MPS

```

1 def otimizar_hiperparametros_qaoa(matriz_w, n_nos):
2     space = [Integer(1, 10, name='p_reps')]
3
4     @use_named_args(space)
5     def objetivo_qaoa(p_reps):
6         corte, _, _ = rodar_qaoa_qiskit(matriz_w, n_nos,
7                                         p_reps=int(p_reps))
8         return -corte
9
10    res = gp_minimize(
11        objetivo_qaoa, space, n_calls=15, random_state=42,
12        base_estimator="GP"
13    )
14    return int(res.x[0])
15
16 def rodar_vqe_qiskit(matriz_w, n_qubits,
17                     multiplicador_iteracoes=100, seed=123):
18     t0 = time.perf_counter()
19     max_cut = Maxcut(matriz_w)
20     qp = max_cut.to_quadratic_program()
21     qubitOp, _ = qp.to_ising()
22
23     limite_iteracoes = max(1000, multiplicador_iteracoes *
24                             (n_qubits ** 2))
25     optimizer = COBYLA(maxiter=limite_iteracoes)

```

```

22
23 # Ansatz TwoLocal (VQE)
24 ansatz_bruto = TwoLocal(qubitOp.num_qubits, "ry", "cz",
25                          reps=3, entanglement="linear")
26
27 backend = AerSimulator()
28 pm = generate_preset_pass_manager(optimization_level=1,
29                                   backend=backend)
30 ansatz_isa = pm.run(ansatz_bruto)
31
32 sampler = AerSampler()
33 sampler.options.default_shots = 1024
34 sampler.options.simulator = {"method": "
35                               matrix_product_state", "seed_simulator": seed}
36
37 vqe = SamplingVQE(sampler=sampler, ansatz=ansatz_isa,
38                  optimizer=optimizer)
39 result = vqe.compute_minimum_eigenvalue(qubitOp)
40
41 x_vqe = max_cut.sample_most_likely(result.eigenstate)
42 corte_vqe = qp.objective.evaluate(x_vqe)
43
44 return corte_vqe, time.perf_counter() - t0, result.
45     cost_function_evals
46
47
48 def rodar_qaoa_qiskit(matriz_w, n_qubits, p_reps=1,
49                       multiplicador_iteracoes=100, seed=123):
50     t0 = time.perf_counter()
51     max_cut = Maxcut(matriz_w)
52     qp = max_cut.to_quadratic_program()
53     qubitOp, _ = qp.to_ising()

```

```

48     limite_iteracoes = max(1000, multiplicador_iteracoes *
49         (n_qubits ** 2))
50
51     # Construção dinamica do circuito QAOA
52     qc = QuantumCircuit(n_qubits)
53     qc.h(range(n_qubits))
54
55     for rep in range(p_reps):
56         gamma = Parameter(f'g_{rep}')
57         beta = Parameter(f'b_{rep}')
58         for i in range(n_qubits):
59             for j in range(i+1, n_qubits):
60                 if matriz_w[i][j] != 0:
61                     qc.cx(i, j)
62                     qc.rz(2.0 * matriz_w[i][j] * gamma, j)
63                     qc.cx(i, j)
64                 qc.rx(2.0 * beta, range(n_qubits))
65
66     backend = AerSimulator()
67     pm = generate_preset_pass_manager(optimization_level=0,
68         backend=backend)
69
70     qc_isa = pm.run(qc)
71
72     sampler = AerSampler()
73     sampler.options.default_shots = 1024
74     sampler.options.simulator = {"method": "
75         matrix_product_state", "seed_simulator": seed}
76
77     qaoa_vqe = SamplingVQE(sampler=sampler, ansatz=qc_isa,
78         optimizer=optimizer)
79     result = qaoa_vqe.compute_minimum_eigenvalue(qubitOp)

```


76

77

```
x_qaoa = max_cut.sample_most_likely(result.eigenstate)
```

78

```
return qp.objective.evaluate(x_qaoa), time.perf_counter  
      () - t0, result.cost_function_evals
```