UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus Bauru

LUCAS ALEXANDRE MARQUES RATTIGHIERI

Aprimoramento de Algoritmos de Feedback Quântico Baseados no Controle de Lyapunov para Preparação de Estados Quânticos

BAURU
2025

LUCAS ALEXANDRE MARQUES RATTIGHIERI

Aprimoramento de Algoritmos de Feedback Quântico Baseados no Controle de Lyapunov para Preparação de Estados Quânticos

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, da Faculdade de Ciências da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de Bauru.

Financiadora: CAPES - Proc. 88887.829793/2023-00

Orientador: Prof. Dr. Felipe Fernandes Fanchini

BAURU

2025

R237a

Rattighieri, Lucas Alexandre Marques
Aprimoramento de Algoritmos de Feedback Quântico Baseados no Controle de Lyapunov para Preparação de Estados Quânticos / Lucas Alexandre Marques Rattighieri. -- Bauru, 2025
102 p.

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru
Orientador: Felipe Fernandes Fanchini

1. Computação Quântica. 2. Algoritmo Quântico Baseado em Feedback. 3. Otimização Quântica. 4. Modelo ANNNI. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Bauru

ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE LUCAS ALEXANDRE MARQUES RATTIGHIERI, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO, DA FACULDADE DE CIÊNCIAS - CÂMPUS DE BAURU.

Aos 09 dias do mês de maio do ano de 2025, às 14h, no(a) Videoconferência, realizou-se a defesa de DISSERTAÇÃO DE MESTRADO de LUCAS ALEXANDRE MARQUES RATTIGHIERI, intitulada **Aprimoramento de Algoritmos de Feedback Quântico Baseados no Controle de Lyapunov para Preparação de Estados Quânticos**. A Comissão Examinadora foi constituída pelos seguintes membros: Prof. Dr. FELIPE FERNANDES FANCHINI (Orientador(a) - Participação Virtual) do(a) Departamento de Física / Faculdade de Ciências UnespCâmpus de Bauru, Prof. Dr. KELTON AUGUSTO PONTARA DA COSTA (Participação Virtual) do(a) FC / UNESPBauru SP, Prof.Dr. LEONARDO KLEBER CASTELANO (Participação Virtual) do(a) Departamento de Física / Universidade Federal de São Carlos. Após a exposição pelo mestrando e arguição pelos membros da Comissão Examinadora que participaram do ato, de forma presencial e/ou virtual, o discente recebeu o conceito final APROVADO . Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelo(a) Presidente(a) da Comissão Examinadora.

Prof. Dr. FELIPE FERNANDES FANCHINI

Dedico este trabalho à minha família.

Agradecimentos

Primeiramente, agradeço a Deus por sua graça e misericórdia sobre minha vida, que me permitiram chegar até aqui.

Agradeço também à minha família, em especial aos meus pais, Rubens e Daniela, pela dedicação e amor, e aos meus irmãos, Beatriz, Filipe e Victor, pelo apoio em todos os momentos.

Sou grato ao professor Felipe Fernandes Fanchini pelos ensinamentos, pelas correções e pela orientação durante a construção deste trabalho.

Aos meus amigos, Jessica, Lucas, Jean e José, agradeço pela companhia e pelo apoio ao longo desta caminhada. Em especial, agradeço ao Guilherme Pexe pela amizade e pela parceria nas pesquisas durante o mestrado.

Por fim, agradeço à Capes pelo suporte financeiro concedido. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

"Aquele que não conheceu pecado, Deus o fez pecado por nós,
para que, nele, fôssemos feitos justiça de Deus."
(Bíblia Sagrada, 2 Coríntios 5, 21)

Resumo

Esta dissertação investiga o aprimoramento do Algoritmo Quântico Baseado em Feedback (FQA) por meio de três frentes principais: o estudo das simetrias do Hamiltoniano, a redução da profundidade dos circuitos via agrupamento de camadas e o reescalonamento temporal da dinâmica. Na primeira frente, analisou-se como as simetrias do Hamiltoniano influenciam a evolução do algoritmo, destacando a preservação de setores simétricos e a importância da escolha do estado inicial. Essa análise foi realizada no modelo ANNNI, que apresenta diferentes simetrias e uma estrutura espectral rica. Na segunda frente, foi desenvolvida a abordagem LGA-FQA, que combina múltiplas camadas em uma única camada efetiva, reduzindo a profundidade dos circuitos e seus custos computacionais. Na terceira, introduziu-se o TR-FQA, que modifica a evolução temporal do algoritmo por meio de uma função de reescalonamento, acelerando a convergência e permitindo o uso de circuitos mais rasos. As abordagens LGA-FQA e TR-FQA foram avaliadas tanto no modelo ANNNI quanto no problema de otimização MaxCut. Os resultados evidenciam que essas abordagens mitigam limitações relacionadas à profundidade de circuitos e à degenerescência espectral, ampliando as possibilidades de aplicação dos algoritmos de feedback quântico.

Palavras-chave: Computação Quântica, Algoritmo Quântico Baseado em Feedback, Redução de Profundidade de Circuitos Quânticos.

Abstract

This dissertation investigates the improvement of the Feedback-Based Quantum Algorithm (FQA) through three main approaches: the study of Hamiltonian symmetries, circuit depth reduction via layer grouping, and temporal rescaling of the algorithm's dynamics. The first approach analyzes how the symmetries of the Hamiltonian influence the algorithm's evolution, highlighting the preservation of symmetric sectors and the importance of the initial state's choice. This analysis was conducted using the ANNNI model, which features distinct symmetries and a complex spectral structure. In the second approach, the LGA-FQA strategy was developed, grouping multiple layers into a single effective layer, thereby reducing circuit depth and computational cost. In the third, the TR-FQA was introduced, modifying the temporal evolution of the algorithm through a rescaling function that accelerates convergence and enables shallower circuits. Both LGA-FQA and TR-FQA were evaluated in the ANNNI model and the MaxCut optimization problem. The results show that these approaches mitigate limitations related to circuit depth and spectral degeneracy, expanding the applicability of feedback-based quantum algorithms.

Keywords: Quantum Computing, Feedback-Based Quantum Algorithm, Quantum Circuit Depth Reduction.

Lista de ilustrações

Figura 1 – Diagrama comparando algoritmos quânticos variacionais (VQAs) e algoritmos quânticos baseados em feedback (FQAs). Ambos podem ser formulados com a mesma função objetivo J e utilizar circuitos quânticos com estrutura idêntica (centro). A diferença está na forma como minimizam J: VQAs ajustam todos os parâmetros $\vec{\theta}$ simultaneamente via otimização clássica (esquerda), enquanto FQAs eliminam essa etapa e definem os parâmetros iterativamente por meio de uma lei de feedback, camada por camada (direita). Figura adaptada de (1). 25

Figura 2 – Diagrama ilustrativo de um FQA. O processo começa com um estado inicial $|\psi_0\rangle$. Em cada camada k, os operadores unitários $e^{-iH_p\Delta t}$ e $e^{-iH_d\Delta t\beta_k}$ são aplicados sequencialmente ao estado atual, ajustando de forma adaptativa o parâmetro β_k. Esse processo é repetido iterativamente para aproximar o estado alvo, representando a evolução do estado $|\psi\rangle$ através das camadas até alcançar a solução desejada. 27

Figura 3 – Diagrama ilustrativo de um WFQAE. O algoritmo começa com a escolha do passo de tempo Δt e do número de camadas ℓ. Em seguida, $p+1$ circuitos quânticos são preparados, cada um inicializado em um estado $\left|\phi_0^{(q)}\right\rangle$, que é fácil de preparar e ortogonal aos estados dos outros circuitos. A construção das camadas ocorre iterativamente, com a aplicação dos operadores $U_d(\beta_k)U_p$ em cada circuito, resultando nos estados $\left|\phi_k^{(q)}\right\rangle$. A configuração final minimiza a função de Lyapunov $J(t)$ associando os estados de menores valores esperados aos maiores pesos, otimizando a convergência para os autovalores do Hamiltoniano. Figura adaptada de (2). 34

Figura 4 – Ilustração da partição de um grafo no problema MaxCut, onde as arestas entre vértices de subconjuntos distintos compõem o corte. Figura elaborada pelo autor. 39

Figura 5 – Representação esquemática do diagrama de fases do modelo ANNNI sob campo transverso. Adaptado de (3, 4, 5). 46

Figura 6 – Representação bloco-diagonal do Hamiltoniano do modelo ANNNI para uma cadeia de 4 sítios, construída a partir dos autovetores comuns à translação e à inversão. Em cada bloco, são indicados os números quânticos p_T e p_I que o caracterizam. Nos setores com número quântico de translação 0 e 2, o Hamiltoniano foi ainda decomposto em blocos menores, utilizando os autoestados comuns à reflexão, sendo os blocos identificados também pelos números quânticos p_R. 51

Figura 7 – Diagrama ilustrativo do LGA-FQA-1. O algoritmo inicia com o estado $|\psi_0\rangle$ e aplica, em cada iteração, a mesma camada unitária composta por $U_d(\alpha_k)$ e $U_p(\delta_k)$, com parâmetros α_k e δ_k atualizados adaptativamente. A cada bloco de n iterações, uma nova camada é inserida no circuito, enquanto as camadas anteriores permanecem fixas. 56
Figura 8 – Diagrama ilustrativo do LGA-FQA-2. O algoritmo começa com o estado inicial $|\psi_0\rangle$ e, em cada iteração, aplica uma camada composta pelos operadores $U_d(\alpha_k)$, $U_p(\delta_k)$ e $U_{com}(\gamma_k)$, com os parâmetros α_k, δ_k e γ_k atualizados de forma adaptativa. A cada bloco de n iterações, uma nova camada agrupada é inserida, enquanto as camadas anteriores permanecem fixas com os últimos parâmetros que possuíram. 57
Figura 9 – Diagrama ilustrativo do TR-FQA. O processo começa com o estado inicial $|\psi_0\rangle$. Em cada camada k, os operadores unitários $e^{-iH_p\dot{f}(k\Delta\tau)\Delta\tau}$ e $e^{-iH_d\dot{f}(k\Delta\tau)\Delta\tau\tilde{\beta}_k}$ são aplicados sequencialmente, ajustando-se de forma adaptativa o parâmetro $\tilde{\beta}_k$. 62
Figura 10 – Convergência do FQA a partir de diferentes estados iniciais pertencentes a setores simétricos distintos no modelo ANNNI com 4 sítios. Cada painel corresponde a um setor de simetria diferente, e a linha tracejada indica a menor energia permitida dentro daquele setor 69
Figura 11 – Convergência do valor esperado da energia $\langle H\rangle_k$ durante a evolução pelo FQA para três estados iniciais distintos: o estado fundamental de $-H_d$, o primeiro estado excitado com simetria compatível com a do primeiro excitado de H_p, e a superposição linear normalizada de ambos. As simulações foram feitas para o modelo ANNNI com diferentes conjuntos de parâmetros κ e g indicados em cada painel, representando três fases distintas do sistema. . . 71
Figura 12 – Convergência do valor esperado da energia ao longo da evolução do WFQAE para diferentes estados iniciais, organizados por setores de simetria. Cada painel refere-se a um setor distinto: (a) $(p_I, p_T, p_R) = (0, 0, 0)$, (b) $(1, 1)$, (c) $(1, 2)$, e (d) $(1, 0, 0)$. Em cada caso, os estados iniciais foram escolhidos da base do respectivo setor, com pesos distintos atribuídos na lei de *feedback*. Observa-se que cada trajetória converge para um autovalor do Hamiltoniano dentro do bloco correspondente ao setor de simetria, evidenciando a capacidade do WFQAE em explorar eficientemente o espectro restrito a cada subespaço. 73

Figura 13 – Simulações numéricas comparando o desempenho dos algoritmos LGA-FQA e FQA. O painel (a) mostra a convergência da função de custo $J = \langle\psi_k|H|\psi_k\rangle$ em função da profundidade relativa do circuito, d_k/d_p, onde d_p corresponde à profundidade de uma camada padrão do FQA. A linha tracejada indica a energia do estado fundamental do Hamiltoniano H_p. O painel (b) apresenta a probabilidade de obtenção de uma solução correta do problema para diferentes profundidades. 75
Figura 14 – Comparação entre os algoritmos FQA e LGA-FQA para diferentes valores do passo de tempo Δt, com $n = 10$ camadas agrupadas. Os painéis (a) e (c) apresentam a convergência da função de custo $J = \langle\psi_k|H|\psi_k\rangle$ em função da profundidade relativa do circuito d_k/d_p, enquanto os painéis (b) e (d) mostram a probabilidade de se obter uma solução correta para diferentes profundidades. Os painéis (a) e (b) referem-se ao LGA-FQA-1 (primeira ordem) e os painéis (c) e (d) ao LGA-FQA-2 (segunda ordem). A linha tracejada representa a energia do estado fundamental de H_p. 76
Figura 15 – Comparação entre os algoritmos FQA e LGA-FQA para diferentes números de camadas agrupadas, com passo de tempo fixo $\Delta t = 0{,}02$. Os painéis (a) e (c) apresentam a convergência da função de custo $J = \langle\psi_k|H|\psi_k\rangle$ em função da profundidade relativa do circuito d_k/d_p, enquanto os painéis (b) e (d) mostram a probabilidade de se obter uma solução correta para diferentes profundidades. Os painéis (a) e (b) referem-se ao LGA-FQA-1 (primeira ordem) e os painéis (c) e (d) ao LGA-FQA-2 (segunda ordem). A linha tracejada representa a energia do estado fundamental de H_p. 78
Figura 16 – Simulações numéricas comparando FQA e LGA-FQA aplicados ao modelo ANNNI, com duas configurações de parâmetros para uma cadeia de 8 sítios: (a) $\kappa = 0.2, g = 0.2$ e (b) $\kappa = 0.8, g = 0.2$. As simulações com LGA-FQA utilizaram a abordagem de agrupamento, com n e Δt indicados em cada painel. O FQA foi simulado com $\Delta t = 0.01$. As linhas tracejadas representam a energia do estado fundamental, enquanto as curvas mostram a convergência da função de custo $J = \langle\Psi_k|H|\Psi_k\rangle$ em função da profundidade relativa d_k/d_p. 80
Figura 17 – Comparação do desempenho do FALQON e TR-FALQON para diferentes funções de reescalonamento temporal. O painel (a) exibe a convergência da função objetivo $\langle\psi_k|H|\psi_k\rangle$ em função do número de camadas k, com a linha tracejada indicando a energia do estado fundamental. O painel (b) apresenta as probabilidades de obter uma solução do problema em certas camadas. 81

Figura 18 – Comparação do desempenho do FALQON e TR-FALQON para diferentes valores do parâmetro de controle de contração temporal a. O painel (a) exibe a convergência da função objetivo $\langle\psi_k|H|\psi_k\rangle$ em função do número de camadas k, com a linha tracejada indicando a energia do estado fundamental. O painel (b) apresenta as probabilidades de obter uma solução do problema em certas camadas. 82
Figura 19 – Comparação do desempenho do FALQON e TR-FALQON para diferentes valores do tempo final t_f. O painel (a) exibe a convergência da função objetivo $\langle\psi_k|H|\psi_k\rangle$ em função do número de camadas k, com a linha tracejada indicando a energia do estado fundamental. O painel (b) apresenta as probabilidades de obter uma solução do problema em certas camadas. . 83
Figura 20 – Simulações numéricas comparando FQA e TR-FQA aplicados ao modelo ANNNI, para diferentes configurações de cadeia e valores de κ e g. As simulações com TR-FQA utilizaram a função de reescalonamento f_1, com os valores de a e t_f indicados em cada painel. Para cadeias com $L = 8$ sítios, adotou-se $\Delta t = \Delta\tau = 0.01$, enquanto para $L = 12$ sítios, usou-se $\Delta t = \Delta\tau = 0.005$. As linhas tracejadas indicam a energia do estado fundamental, enquanto as curvas mostram a convergência da função de custo $J = \langle\Psi_k|H|\Psi_k\rangle$ em função do número de camadas k. 84

Lista de abreviaturas e siglas

VQA	Variational Quantum Algorithm
VQE	Variational Quantum Eigensolver
QAOA	Quantum Approximate Optimization Algorithm
FQA	Feedback-Based Quantum Algorithm
FALQON	Feedback-Based Algorithm for Quantum Optimization
NISQ	Noisy Intermediate-Scale Quantum
ANNNI	Axial Next-Nearest-Neighbor Ising Model
LGA	Layer Grouping Approach
TR-FQA	Time-Rescaled Feedback Quantum Algorithm
QLC	Quantum Lyapunov Control
WFQAE	Weighted Feedback-Based Quantum Algorithm for Excited States Calculation
QUBO	Quadratic Unconstrained Binary Optimization
BCH	Identidade de Baker-Campbell-Hausdorff

Sumário

1 INTRODUÇÃO . . . 16
1.1 Objetivos . . . 18
1.1.1 Objetivo Geral . . . 18
1.1.2 Objetivos Específicos . . . 18

2 CONTROLE DE LIAPUNOV QUÂNTICO . . . 19
2.0.1 Funções de Lyapunov alternativas . . . 20
2.1 Inclusão de uma Perturbação de Referência em $\beta(t)$ . . . 21
2.2 Controle Iterativo de Lyapunov Quântico . . . 22
2.3 Múltiplas Funções de Controle . . . 23

3 FEEDBACK-BASED QUANTUM ALGORITHMS . . . 25
3.1 Construção do Circuito no FQA a partir da Dinâmica de Evolução Temporal . . . 26
3.2 Escolha de Δt . . . 28
3.3 Variações de FQAs . . . 29
3.3.1 FQA com Perturbações de Referência . . . 29
3.3.2 FQA com Função de Controle Iterativa . . . 29
3.3.3 FQA com Múltiplas Funções de Controle . . . 29
3.3.4 SO-FALQON . . . 30
3.3.5 WFQAE . . . 31

4 CODIFICAÇÃO DE PROBLEMAS DE OTIMIZAÇÃO PARA ALGORITMOS QUÂNTICOS . . . 35
4.1 Otimização Binária Quadrática Irrestrita . . . 35
4.1.1 Codificação de Restrições . . . 35
4.2 Transcrição de um Problema QUBO para um Hamiltoniano . . . 37
4.3 MaxCut . . . 39

5 SIMETRIAS . . . 41
5.1 Conservação dos números quânticos na evolução temporal . . . 42
5.2 Simetrias Cíclicas . . . 43

6 MODELO ANNNI . . . 46
6.1 Simetrias do Modelo ANNNI . . . 47

7 AGRUPAMENTOS DE CAMADAS EM FQAS . . . 52

7.1 **Agrupamento de Camadas com Aproximação de Primeira Ordem** . . **52**
7.2 **Agrupamento de Camadas com Aproximação de Segunda Ordem** . **53**
7.3 **LGA-FQA** **55**
7.3.1 LGA-FQA-1 55
7.3.2 LGA-FQA-2 56
7.4 **Comparação entre Profundidades** **58**

8 **ALGORITMO QUÂNTICO BASEADO EM FEEDBACK COM RE-ESCALONAMENTO TEMPORAL** **60**
8.1 **Método de Reescalonamento Temporal** **60**
8.2 **TR-FQA** **61**

9 **METODOLOGIA** **64**
9.1 **Influência das Simetrias no Desempenho do FQA** **65**
9.2 **Impacto do Agrupamento de Camadas** **66**
9.3 **Efeito do Reescalonamento Temporal na Convergência do FQA** . . **67**

10 **RESULTADOS** **69**
10.1 **Efeitos das Simetrias na Convergência do FQA** **69**
10.2 **LGA-FQA e LGA-FALQON** **74**
10.3 **TR-FQA e TR-FALQON** **80**

11 **CONCLUSÃO** **86**

REFERÊNCIAS **88**

APÊNDICES **93**

APÊNDICE A – FÓRMULA DE TROTTER **94**

APÊNDICE B – IDENTIDADE DE BAKER-CAMPBELL-HAUSDORFF (BCH) **97**

1 Introdução

A computação quântica tem se destacado nos últimos anos devido ao seu potencial para resolver problemas ineficientes para computadores clássicos, como a simulação de sistemas quânticos e a solução de problemas de otimização. Com o aumento do número de parâmetros desses problemas, explorar o espaço de busca torna-se uma tarefa árdua para os computadores atuais, o que cria oportunidades para o uso de computadores quânticos como uma alternativa viável.

Entre os paradigmas da computação quântica, o modelo de portas quânticas é um dos mais estudados. Nesse modelo, algoritmos quânticos são construídos a partir de operações discretas implementadas por portas quânticas, que realizam transformações unitárias sobre o estado de entrada. Essas transformações evoluem o sistema de forma controlada, de modo que, ao final do circuito, o estado quântico resultante apresenta amplitudes associadas às possíveis saídas do problema. A informação desejada está codificada nessas amplitudes e pode ser acessada por meio da repetição de medições sobre o circuito, permitindo estimar a distribuição de probabilidades associada aos resultados. Algoritmos como os de Deutsch-Jozsa (6) e Grover (7) utilizam esse paradigma para resolver problemas específicos, apresentando um ganho de eficiência em comparação com soluções clássicas.

Entretanto, sua implementação prática ainda enfrenta diversos desafios, principalmente em relação à profundidade dos circuitos necessários para a preparação e execução dos estados. A profundidade maior do circuito acarreta uma maior suscetibilidade a erros, representando uma limitação nos dispositivos de escala intermediária ruidosa (NISQ) (8, 9), que ainda apresentam restrições de fidelidade e tempo de coerência. Esses obstáculos dificultam a execução de algoritmos quânticos nesses dispositivos, demandando estratégias que aumentem a eficiência e reduzam os custos computacionais.

Entre os algoritmos quânticos relevantes, destacam-se os algoritmos quânticos variacionais (VQAs) (10, 11), como o *Variational Quantum Eigensolver* (VQE) (12, 13, 14, 15, 16, 17, 18) e o *Quantum Approximate Optimization Algorithm* (QAOA) (19, 20, 21, 22), que combinam circuitos quânticos parametrizados com otimizadores clássicos. Embora esses métodos tenham mostrado sucesso em vários contextos, como a resolução de problemas de otimização e simulação de sistemas quânticos, eles dependem de uma otimização clássica para ajustar os parâmetros dos circuitos quânticos. A otimização dos parâmetros, entretanto, pode ser tão complexa quanto o próprio problema original, especialmente à medida que o espaço de busca se expande (23).

Como alternativa a esses métodos híbridos, os algoritmos baseados em feedback quântico (24, 25, 1), como o *Feedback-Based Quantum Algorithm* (FQA) e sua especialização para

otimização, o *Feedback-Based Algorithm for Quantum Optimization* (FALQON), oferecem uma solução sem a necessidade de otimização clássica. Nesse tipo de abordagem, o circuito quântico é construído de forma iterativa, camada por camada. A cada etapa, realiza-se uma medição no estado gerado pela camada atual, e o resultado é utilizado para ajustar os parâmetros da próxima camada. Esse mecanismo de feedback direciona o sistema ao longo de sua evolução, com o objetivo de aproximar-se do estado desejado, geralmente o estado fundamental do Hamiltoniano, sem recorrer a otimizadores clássicos.

Apesar dessa vantagem, sua principal limitação está na profundidade dos circuitos necessária para alcançar bons resultados. Frequentemente, são exigidas muitas camadas para garantir a convergência, o que inviabiliza a execução desses algoritmos em dispositivos com recursos limitados. Diversas estratégias têm sido exploradas para contornar essa limitação, como a modificação das regras de feedback e a reorganização da estrutura dos circuitos para reduzir a profundidade total.

Este trabalho propõe duas abordagens complementares para mitigar esse problema nos algoritmos FQA e FALQON: o *Layer Grouping Approach* (LGA-FQA) e o *Time-Rescaled Feedback Quantum Algorithm* (TR-FQA). A primeira abordagem agrupa múltiplas camadas em uma única camada efetiva, reduzindo a profundidade do circuito sem alterar a estrutura do algoritmo. A segunda modifica a dinâmica do Hamiltoniano de referência via reescalonamento temporal, acelerando a evolução do sistema e permitindo a execução com menos camadas.

Além dessas propostas, investiga-se o papel das simetrias do Hamiltoniano no desempenho dos algoritmos baseados em feedback. O modelo ANNNI (*Axial Next-Nearest-Neighbor Ising*) é utilizado como caso de estudo para examinar como essas simetrias influenciam a convergência e a eficiência algorítmica. Os métodos LGA-FQA e TR-FQA são testados nesse modelo para avaliar sua eficácia na preparação de estados físicos e também aplicados ao problema de otimização MaxCut, onde o objetivo é maximizar a probabilidade de medir a solução correta, e não necessariamente atingir o estado fundamental.

Com essas abordagens, esta dissertação busca contribuir para a execução mais eficiente de algoritmos quânticos baseados em feedback, tornando-os mais viáveis em dispositivos NISQ. O estudo das simetrias e das estratégias de redução de profundidade visa aprofundar o entendimento sobre o funcionamento desses algoritmos, identificar suas limitações e explorar sua aplicabilidade, tanto em simulações de sistemas físicos quanto em problemas de otimização combinatória.

1.1 Objetivos

1.1.1 Objetivo Geral

Investigar e propor estratégias para reduzir a profundidade dos circuitos em algoritmos quânticos baseados em feedback, além de analisar o papel das simetrias do Hamiltoniano no desempenho desses algoritmos em sistemas físicos, e avaliar sua aplicação em problemas de otimização.

1.1.2 Objetivos Específicos

- Estudar o impacto das simetrias do Hamiltoniano sobre a dinâmica do algoritmo FQA, com aplicação ao modelo ANNNI;
- Desenvolver e implementar a abordagem LGA-FQA, baseada no agrupamento de camadas, visando a redução da profundidade dos circuitos;
- Formular o algoritmo TR-FQA, utilizando reescalonamento temporal para acelerar a convergência e reduzir a profundidade dos circuitos;
- Aplicar as estratégias propostas ao problema de otimização MaxCut, avaliando sua eficácia na aproximação de soluções ótimas;
- Comparar o desempenho das abordagens LGA-FQA e TR-FQA com os algoritmos originais FQA e FALQON.

2 Controle de Liapunov Quântico

O Controle de Lyapunov Quântico (*Quantum Lyapunov Control* - QLC) (26, 25, 27) é um método utilizado para identificar controles que guiem assintoticamente a evolução de um sistema quântico para um estado alvo. Para compreender o QLC, começamos considerando um sistema governado pelo Hamiltoniano $H(t) = H_p + \beta(t)H_d$, cuja dinâmica é regida pela equação de Schrödinger dependente do tempo ($\hbar = 1$):

$$i\frac{d}{dt}\left|\psi(t)\right\rangle = \left(H_p + \beta(t)H_d\right)\left|\psi(t)\right\rangle \tag{1}$$

onde $|\psi(t)\rangle$ representa o vetor de estado do sistema no instante t. Os operadores H_p e H_d são Hamiltonianos adimensionais: H_p corresponde ao Hamiltoniano do problema, que codifica a estrutura do problema a ser resolvido, enquanto H_d, chamado de Hamiltoniano de condução ou controle, é responsável por conduzir a evolução do sistema. A função de controle escalar dependente do tempo é representada por $\beta(t)$, acoplada a H_d. O objetivo do controle é determinar a função $\beta(t)$ que conduza assintoticamente o estado do sistema $|\psi(t)\rangle$ a um estado que minimize uma função objetivo J, conhecida como função de Lyapunov. Neste caso, J é definida como o valor esperado de H_p:

$$J(t) = \left\langle\psi(t)\right| H_p \left|\psi(t)\right\rangle . \tag{2}$$

Minimizando J, obtemos o estado fundamental deste hamiltoniano. O QLC ajustará $\beta(t)$ para garantir que:

$$\frac{dJ}{dt}(t) \leq 0, \forall t \tag{3}$$

Ao calcular a derivada de J em relação a t, obtemos:

$$\begin{aligned}
\frac{dJ}{dt}(t) &= \frac{d}{dt}\left\langle\psi(t)\right| H_p \left|\psi(t)\right\rangle \\
&= i\left\langle\psi(t)\right| H(t)H_p \left|\psi(t)\right\rangle - i\left\langle\psi(t)\right| H_pH(t) \left|\psi(t)\right\rangle \\
&= \left\langle\psi(t)\right| i(H(t)H_p - H(t)H_d)\beta(t) \left|\psi(t)\right\rangle \\
&= \left\langle\psi(t)\right| i[H(t), H_p] \left|\psi(t)\right\rangle \beta(t) \\
&= \left\langle\psi(t)\right| i[H_p + \beta(t)H_d, H_p] \left|\psi(t)\right\rangle \beta(t) \\
&= \left\langle\psi(t)\right| i[H_d, H_p] \left|\psi(t)\right\rangle \beta(t)
\end{aligned} \tag{4}$$

onde a Eq. (1) foi utilizada para obter este resultado. Definimos $A(t) = \left\langle\psi(t)\right| i[H_d, H_p] \left|\psi(t)\right\rangle$, de forma que:

$$\frac{dJ}{dt}(t) = A(t)\beta(t) \tag{5}$$

Para que a condição (3) seja satisfeita, deve-se ter:

$$\beta(t) = -\omega f(t, A(t)) \tag{6}$$

onde $\omega > 0$ e $f(t, A(t))$ é uma função contínua com $f(t, 0) = 0$ e $A(t)f(t, A(t)) > 0$ para todo $A(t) \neq 0$. A escolha de $\beta(t)$ dessa forma garante que a função $J(t)$ diminua ao longo do tempo.

A Eq. (6) oferece flexibilidade na escolha de $\beta(t)$. A escolha mais simples e comum é considerar $\omega = 1$ e $f(t, A(t)) = A(t)$, resultando em:

$$\beta(t) = -A(t) \tag{7}$$

o que satisfaz a condição (3), pois $\frac{dJ}{dt} = -(A(t))^2 \leq 0$. É importante destacar que H_p e H_d não devem comutar, pois, caso comutem, teríamos $A(t) = 0$ para todo t, impedindo o decréscimo da energia ao longo do tempo.

2.0.1 Funções de Lyapunov alternativas

Além da função de Lyapunov baseada no valor esperado de H_p, outras formas de funções de Lyapunov podem ser empregadas, dependendo do objetivo e das informações disponíveis sobre o sistema (28). Uma alternativa comum é a função baseada na distância entre estados, definida como:

$$J(t) = \frac{1}{2}\left(1 - |\langle\psi(t)|\psi_f\rangle|^2\right) \tag{8}$$

onde $|\psi_f\rangle$ é o estado alvo. O termo $|\langle\psi(t)|\psi_f\rangle|^2$ representa a fidelidade entre o estado atual $|\psi(t)\rangle$ e o estado alvo $|\psi_f\rangle$, assumindo valores entre 0 e 1. Assim, $J(t)$ mede quão distante o sistema está de $|\psi_f\rangle$, atingindo o valor mínimo zero somente quando $|\psi(t)\rangle = e^{i\phi}|\psi_f\rangle$, ou seja, a menos de uma fase global.

Para que essa função leve de fato à convergência do sistema para o estado alvo, é necessário que $|\psi_f\rangle$ seja um autoestado do Hamiltoniano H_p. Isso ocorre porque, como a dinâmica livre é regida por H_p, o sistema tende naturalmente a permanecer (ou evoluir ciclicamente) dentro da base de seus autoestados. Se $|\psi_f\rangle$ não for um autoestado, o controle pode competir com a dinâmica natural, dificultando ou até impedindo a convergência.

Outra possibilidade é a função de Lyapunov baseada diretamente no erro de estados, que utiliza a norma da diferença entre os vetores de estado:

$$J(t) = \frac{1}{2}\| \, |\psi(t)\rangle - |\psi_f\rangle \, \|^2 = \frac{1}{2} \left(\langle \psi(t) - \psi_f | \psi(t) - \psi_f \rangle \right). \tag{9}$$

Essa função também é positiva e se anula somente quando $|\psi(t)\rangle = |\psi_f\rangle$ (a menos de uma fase global). Nesses dois casos a função de controle depende de $|\psi_f\rangle$, o que significa que o estado alvo deve ser conhecido previamente, o que pode não ser desejável ou viável em certos cenários, como na busca por estados fundamentais desconhecidos.

Por fim, há também a possibilidade de definir a função de Lyapunov como o valor esperado de um operador P que comuta com H_p, ou seja, tal que $[P, H_p] = 0$:

$$J(t) = \langle \psi(t) | \, P \, | \psi(t) \rangle \tag{10}$$

Nesse caso, como P pode ser diagonalizado na mesma base de H_p, é possível projetar a dinâmica de forma a forçar a convergência para um subespaço associado a autovalores específicos. Contudo, a convergência será para esse subespaço e não necessariamente para um único vetor de estado.

Neste trabalho, utiliza-se o valor esperado de H_p como função de Lyapunov, uma vez que o algoritmo baseado em feedback fundamenta-se nessa função para guiar a evolução do sistema. Por ser simples e diretamente relacionada à minimização da energia, cujo mínimo corresponde ao estado fundamental de H_p, essa escolha se mostra adequada. Além disso, não exige conhecimento prévio do estado alvo, o que a torna útil em problemas de otimização e preparação de estados fundamentais.

2.1 Inclusão de uma Perturbação de Referência em $\beta(t)$

Uma estratégia para aprimorar a convergência ao mínimo global de J é adicionar uma perturbação de referência à função $\beta(t)$ (25). Essa abordagem consiste em incluir uma perturbação de referência, $\lambda(t)$, de forma que a função de controle passe a ser $(\lambda(t) + \beta(t))$. Assim, o hamiltoniano dependente do tempo do sistema é expresso como:

$$H(t) = H_p + (\lambda(t) + \beta(t)) H_d \tag{11}$$

Após a introdução da perturbação, podemos definir um sistema perturbado, denotado por (b), cuja dinâmica é governada pelo hamiltoniano dependente do tempo:

$$H_{p,(b)}(t) = H_p + \lambda(t) H_d \tag{12}$$

Para esse sistema, definimos a função de Lyapunov perturbada como $J_{(b)}(t) = \langle \psi(t) | \, H_{p,(b)} \, | \psi(t) \rangle$. Para garantir a convergência, asseguramos que:

$$\frac{dJ_{(b)}}{dt} \leq 0 \tag{13}$$

Ao analisar essa derivada, obtemos:

$$\begin{aligned}
\frac{dJ_{(b)}}{dt}(t) &= \frac{d}{dt} \langle \psi(t)| H_{p,(b)} |\psi(t)\rangle \\
&= \langle \psi(t)| i[H(t), H_{p,(b)}] |\psi(t)\rangle \\
&= \langle \psi(t)| i[H_p + (\lambda(t) + \beta(t))H_d, H_p + \lambda(t)H_d] |\psi(t)\rangle \\
&= \langle \psi(t)| i\left(-\lambda(t)[H_d, H_p] + (\lambda(t) + \beta(t))[H_d, H_p]\right) |\psi(t)\rangle \\
&= \langle \psi(t)| i[H_d, H_p] |\psi(t)\rangle \beta(t) \\
&= A(t)\beta(t)
\end{aligned} \tag{14}$$

Assim, a inclusão da perturbação não altera a lei de controle, permitindo o uso da mesma função definida na Eq. (6). O objetivo é que essa nova função de Lyapunov melhore a convergência ao mínimo global. Portanto, se o sistema (b) convergir para o estado fundamental de $H_{p,(b)}$ em um tempo T, e $\lambda(t)$ for definida de modo que $\lambda(T) = 0$, o sistema (b) atingirá o estado fundamental de H_p. Dessa forma, $\lambda(t)$ pode ser escolhida como uma função que tenda a zero à medida que o tempo avança.

2.2 Controle Iterativo de Lyapunov Quântico

Outro método para melhorar a convergência de $J(t)$ é o uso de um procedimento iterativo para refinar a função de controle $\beta(t)$. Nesse método, o sistema é inicialmente preparado em um estado $|\psi_0\rangle$ no tempo $t = 0$ e evolui ao longo de um intervalo $[0, T]$. Na primeira iteração, o sistema é guiado pela função de controle $\beta^{(0)}(t)$, definida conforme a Eq. (6), gerando a trajetória $J^{(0)}(t)$. Cada evolução de $t = 0$ até $t = T$ corresponde a uma iteração.

Nas iterações subsequentes, para $j \geq 1$, a função de controle da iteração anterior, $\beta^{(j-1)}(t)$, é tratada como uma perturbação de referência, $\lambda(t) = \beta^{(j-1)}(t)$, conforme descrito na seção 2.1. Com isso, o sistema da iteração anterior passa a ser considerado como um sistema perturbado:

$$H_{p,(b)}^{(j-1)}(t) = H_p + \beta^{(j-1)}(t)H_d \tag{15}$$

Em seguida, uma nova função de controle $\tilde{\beta}^{(j)}(t)$ é determinada para o intervalo de tempo $[0, T]$, conforme a Eq. (6), utilizando a função de Lyapunov $J_{(b)}^{(j)}(t) = \langle \psi(t)| H_p^{(j-1)} |\psi(t)\rangle$. Após essa etapa, o campo de controle é atualizado somando a nova função à função anterior:

$$\beta^{(j)}(t) = \beta^{(j-1)}(t) + \tilde{\beta}^{(j)}(t) \tag{16}$$

Se o tempo T for escolhido adequadamente longo, de modo que $\beta^{(j)}(T) = 0$, fazendo com que $H^{(j)}_{\mathrm{p},(b)}(T) = H_{\mathrm{p}}$, ao final de cada iteração, o sistema estará mais próximo do objetivo em comparação com a iteração anterior:

$$\begin{aligned}
\left\langle \psi^{(j)}(T) \left| H_{\mathrm{p}} \right| \psi^{(j)}(T) \right\rangle &= \left\langle \psi^{(j)}(T) \left| H^{(j-1)}_{\mathrm{p},(b)}(T) \right| \psi^{(j)}(T) \right\rangle \\
&\leq \left\langle \psi^{(j)}(0) \left| H^{(j-1)}_{\mathrm{p},(b)}(0) \right| \psi^{(j)}(0) \right\rangle \\
&= \left\langle \psi_0 \left| H^{(j-1)}_{\mathrm{p},(b)}(0) \right| \psi_0 \right\rangle \\
&= \left\langle \psi^{(j-1)}(0) \left| H^{(j-1)}_{\mathrm{p},(b)}(0) \right| \psi^{(j-1)}(0) \right\rangle \\
&= \left\langle \psi^{(j-1)}(T) \left| H^{(j-1)}_{\mathrm{p},(b)}(T) \right| \psi^{(j-1)}(T) \right\rangle \\
&= \left\langle \psi^{(j-1)}(T) \left| H_{\mathrm{p}} \right| \psi^{(j-1)}(T) \right\rangle
\end{aligned} \tag{17}$$

2.3 Múltiplas Funções de Controle

A estrutura do controle de Lyapunov quântico pode ser ampliada para sistemas que envolvem mais de uma função de controle. Consideremos, para isso, que o Hamiltoniano do sistema seja descrito por:

$$H(t) = H_p + \sum_j \beta(j,t) H_{d,j}, \tag{18}$$

onde $\beta(j,t)$ representa a j-ésima função de controle associada ao Hamiltoniano de condução $H_{d,j}$. Nesse contexto, a função de Lyapunov permanece a mesma, $J(t) = \langle \psi(t) | H_p | \psi(t) \rangle$, devendo satisfazer a condição dada em (3). Assim, temos:

$$\begin{aligned}
\frac{dJ}{dt}(t) &= \frac{d}{dt} \langle \psi(t) | \, H_p \, | \psi(t) \rangle \\
&= \langle \psi(t) | \, i[H(t), H_p] \, | \psi(t) \rangle \\
&= \langle \psi(t) | \, i \left[\sum_j \beta(j,t) H_{d,j}, H_p \right] | \psi(t) \rangle \\
&= \sum_j \langle \psi(t) | \, i[H_{d,j}, H_p] \, | \psi(t) \rangle \, \beta(j,t) \\
&= \sum_j A(j,t) \beta(j,t)
\end{aligned} \tag{19}$$

onde $A(j,t) = \langle \psi(t) | [H_{d,j}, H_p] | \psi(t) \rangle$. Portanto, para cada função de controle:

$$\beta(j,t) = -\omega_j f_j(t, A(j,t)), \tag{20}$$

onde, ω_i e f_i seguem as mesmas condições que ω e f presentes na Eq. (6), garantindo assim que a função de Lyapunov decresça ao longo do tempo.

3 Feedback-based Quantum Algorithms

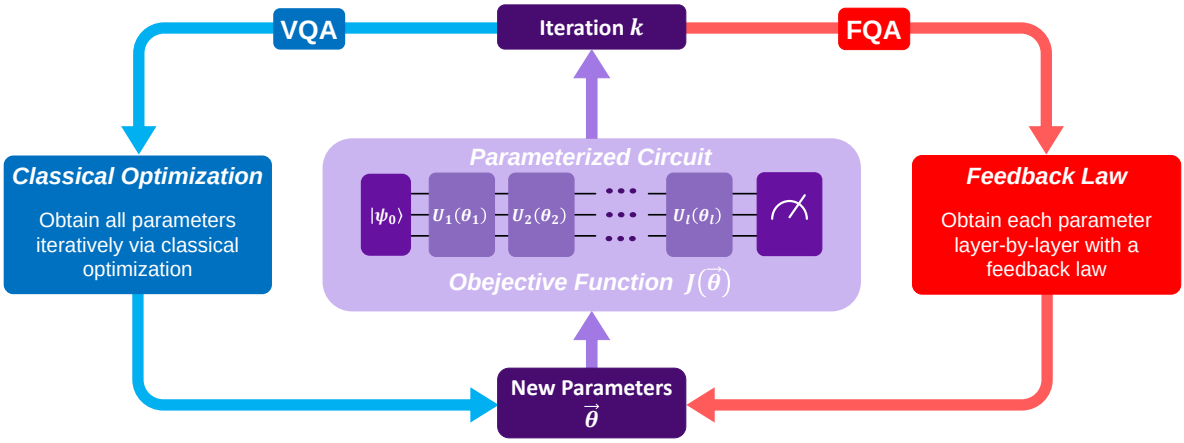

Figura 1 – Diagrama comparando algoritmos quânticos variacionais (VQAs) e algoritmos quânticos baseados em feedback (FQAs). Ambos podem ser formulados com a mesma função objetivo J e utilizar circuitos quânticos com estrutura idêntica (centro). A diferença está na forma como minimizam J: VQAs ajustam todos os parâmetros $\vec{\theta}$ simultaneamente via otimização clássica (esquerda), enquanto FQAs eliminam essa etapa e definem os parâmetros iterativamente por meio de uma lei de feedback, camada por camada (direita). Figura adaptada de (1).

Os Algoritmos Quânticos Baseados em Feedback (*Feedback-based Quantum Algorithms* - FQAs) formam uma classe de algoritmos quânticos projetados para preparar estados quânticos, com foco particular no estado fundamental de um Hamiltoniano alvo. Esses algoritmos se baseiam no controle quântico de Lyapunov e operam de forma iterativa, construindo sucessivamente camadas de um circuito quântico parametrizado.

Por outro lado, os algoritmos quânticos variacionais (*Variational Quantum Algorithms* – VQAs) buscam minimizar uma função custo $J(\vec{\theta})$ utilizando uma abordagem híbrida, combinando computação clássica e quântica. Em um computador quântico, um circuito parametrizado $U(\vec{\theta})$ é aplicado a um estado inicial $|\psi_0\rangle$, resultando no estado $\left|\psi(\vec{\theta})\right\rangle = U(\vec{\theta})\,|\psi_0\rangle$. Esse estado é utilizado para estimar $J(\vec{\theta})$, que geralmente corresponde ao valor esperado de um Hamiltoniano. Em um computador clássico, um algoritmo de otimização ajusta os parâmetros $\vec{\theta}$ com o objetivo de encontrar a configuração $\vec{\theta}_{\text{min}}$ que minimiza J. Durante o processo, a estrutura do circuito quântico permanece fixa; apenas os valores de $\vec{\theta}$ são atualizados a cada iteração.

Os FQAs, por sua vez, eliminam a etapa de otimização clássica e empregam uma abordagem de feedback. A cada iteração, uma nova camada parametrizada é adicionada ao circuito, enquanto as camadas anteriores permanecem fixas. Os parâmetros da nova camada são determinados com base em medições realizadas no estado preparado até aquele ponto, de

acordo com uma lei de feedback derivada da teoria de controle de Lyapunov. Esse mecanismo de realimentação orienta o sistema, passo a passo, em direção ao estado fundamental do Hamiltoniano.

A Figura 1 ilustra graficamente a distinção conceitual entre um VQA e um FQA.

3.1 Construção do Circuito no FQA a partir da Dinâmica de Evolução Temporal

O funcionamento do FQA pode ser compreendido a partir da solução geral da Eq. (1), que assume a forma:

$$|\psi(t)\rangle = \mathcal{T} e^{\int_0^t -i(H_p+\beta(t')H_d)dt'} |\psi(0)\rangle \,, \tag{21}$$

onde $\mathcal{T}$ é o operador de ordenação temporal. Ao discretizar o intervalo de tempo $[0, t]$ em k passos de duração Δt, nos quais o hamiltoniano pode ser considerado como independente do tempo em cada passo, a expressão (21) pode ser aproximada por:

$$|\psi(t)\rangle \approx \mathcal{T} e^{\sum_{k=0}^{t/\Delta t} -i(H_p+\beta(k\Delta t)H_d)\Delta t} |\psi(0)\rangle \,. \tag{22}$$

Essa expressão pode ser interpretada como a aplicação sucessiva de operadores unitários do tipo $U(t) = e^{-i(H_p+\beta(t)H_d)\Delta t}$, que, ao serem aplicados ao estado $|\psi(t)\rangle$, o levam para o estado $|\psi(t+\Delta t)\rangle$, resultando em:

$$|\psi(k\Delta t)\rangle \approx e^{-i(H_p+\beta(k\Delta t)H_d)\Delta t} \dots e^{-i(H_p+\beta(2\Delta t)H_d)\Delta t} e^{-i(H_p+\beta(\Delta t)H_d)\Delta t} |\psi(0)\rangle \,. \tag{23}$$

O próximo passo consiste em simplificar a expressão de $U(t)$, visando sua implementação em circuitos quânticos. Para isso, assume-se que Δt é suficientemente pequeno, o que permite utilizar uma aproximação trotterizada (detalhada no Apêndice A):

$$U(t) = e^{-i(H_p+\beta(t)H_d)\Delta t} \approx e^{-i\beta(t)H_d\Delta t} e^{-iH_p\Delta t} = U_d(\beta(t))U_p, \tag{24}$$

com $U_d(\beta(t)) =^{-i\beta(t)H_d\Delta t}$ e $U_p = e^{-iH_p\Delta t}$.

Isso nos leva à expressão:

$$|\psi_k\rangle = e^{-i\beta_k H_d\Delta t} e^{-iH_p\Delta t} \dots e^{-i\beta_2 H_d\Delta t} e^{-iH_p\Delta t}\, e^{-i\beta_1 H_d\Delta t} e^{-iH_p\Delta t} |\psi_0\rangle \,. \tag{25}$$

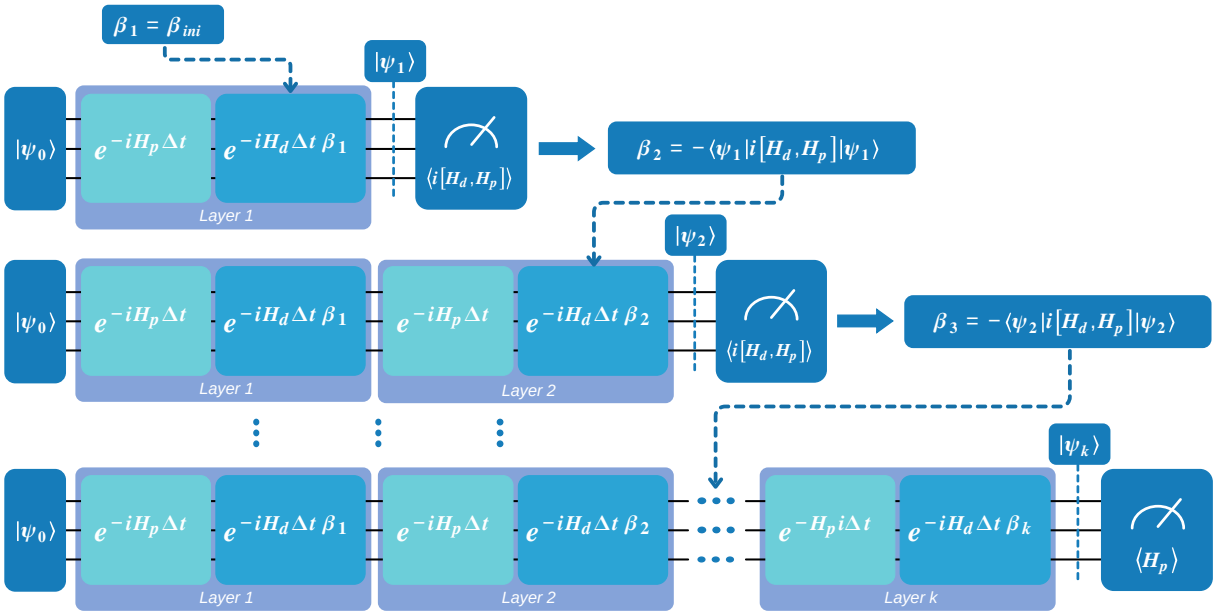

Figura 2 – Diagrama ilustrativo de um FQA. O processo começa com um estado inicial $|\psi_0\rangle$. Em cada camada k, os operadores unitários $e^{-iH_p\Delta t}$ e $e^{-iH_d\Delta t\beta_k}$ são aplicados sequencialmente ao estado atual, ajustando de forma adaptativa o parâmetro β_k. Esse processo é repetido iterativamente para aproximar o estado alvo, representando a evolução do estado $|\psi\rangle$ através das camadas até alcançar a solução desejada.

onde $|\psi_k\rangle$ é o estado do circuito após a aplicação k-ésima camada ($|\psi_k\rangle \approx |\psi(k\Delta t)\rangle$) e β_k é o parâmetro de controle da k-ésima camada ($\beta_k \approx \beta(k\Delta t)$). Para garantir que o sistema evolua em direção ao estado fundamental de H_p, definimos β_k a partir da Eq. (7), onde β em um dado instante depende do estado do sistema naquele momento. No entanto, como precisamos de β_k antecipadamente para construir o estado $|\psi_k\rangle$, utilizamos o estado da iteração anterior, que é muito próximo de $|\psi_k\rangle$ para pequenos intervalos de tempo:

$$\beta_k = -A_{k-1} = -\langle\psi_{k-1}|\, H_p\, |\psi_{k-1}\rangle\,, \tag{26}$$

o que estabelece a Lei de Feedback. Com isso, cada nova camada é construída com base em informações extraídas do estado gerado pela camada anterior.

Dessa forma, o FQA começa com a preparação de um estado inicial $|\psi_0\rangle$ de fácil implementação. Após a escolha do número de camadas ℓ e de um valor apropriado de Δt, aplica-se a primeira camada com $\beta_1 = 0$. Em cada camada subsequente k, os operadores $e^{-iH_p\Delta t}$ e $e^{-iH_d\Delta t\beta_k}$ são aplicados sequencialmente, com β_k determinado pela Eq. (26) a partir do estado $|\psi_{k-1}\rangle$. O nome "feedback"do algoritmo se deve a esse processo, no qual o parâmetro de uma nova camada é determinado a partir da medição do estado gerado pela camada anterior. Esse processo iterativo ajusta o estado do sistema progressivamente, reduzindo o valor da

função de custo $J_k = \langle\psi_k| H_p |\psi_k\rangle$ a cada passo ($J_1 \geq J_2 \geq \cdots \geq J_k$).

A Figura 2 ilustra o funcionamento do algoritmo, evidenciando a estrutura camada por camada e o papel da retroalimentação no ajuste dos parâmetros.

Esse tipo de algoritmo pode ser utilizado para preparar o estado fundamental de um sistema físico de interesse ou um estado específico $|\phi\rangle$, codificado como o estado fundamental do hamiltoniano $H_p = \mathbb{1} - |\phi\rangle\langle\phi|$. Quando aplicado à resolução de problemas de otimização combinatória, o FQA é conhecido como FALQON (*Feedback-based ALgorithm for Quantum OptimizatioN*) (24).

3.2 Escolha de Δt

A escolha do passo de tempo Δt é importante para o bom funcionamento do algoritmo. Um passo muito pequeno resulta em uma evolução lenta, exigindo um maior número de camadas para que o algoritmo converja para o estado alvo. Por outro lado, um passo grande demais pode causar saltos excessivos, comprometendo a suposição de que o hamiltoniano é independente do tempo em cada passo, o que é fundamental para a validade da Eq. (22) e da trottenização que leva à Eq. (25). Além disso, um Δt grande pode tornar $|\psi_{k-1}\rangle$ muito diferente de $|\psi_k\rangle$, inviabilizando a lei de feedback. Isso pode fazer com que a função objetivo não diminua monotonicamente a cada passo, provocando oscilações drásticas tanto nessa função quanto em β. O maior valor de Δt que ainda permite a convergência é conhecido como passo de tempo crítico, Δt_c.

Um limite para Δt, que garante a manutenção dessas condições, é apresentado em (1), derivado a partir dos passos discutidos em (29):

$$\Delta t \leq \frac{1}{4\,\|H_p\|\,\|H_d\|^2} \tag{27}$$

onde $\|\cdot\|$ representa a norma espectral dos hamiltonianos H_p e H_d, definida como o maior valor absoluto entre seus autovalores.

Outro limite para a escolha de Δt é apresentado em (25), baseado na condição de que a energia deve diminuir de um passo para outro. Isso requer que os termos de primeira ordem na expansão dos operadores U_p e $U_d(\beta_k)$ predominem sobre os demais, resultando na seguinte desigualdade:

$$|\Delta t| < \frac{|A_k|}{2\left(2\|H_d\|\|H_p\| + |A_k|\right)\left(\|H_p\| + \|H_d\|\,|\beta_k|\right)} \tag{28}$$

Essa desigualdade permite a escolha de valores adaptativos para Δt, atualizados a cada camada com base nas medições de A_k.

Na prática, é possível adotar valores de Δt maiores que os limites dados pelas Eqs. (27) e (28), desde que estejam abaixo de Δt_c, de forma a acelerar a convergência sem comprometer a estabilidade do algoritmo.

3.3 Variações de FQAs

Nesta seção, serão apresentadas algumas variações dos FQAs existentes que visam melhorar a eficiência e a convergência do algoritmo.

3.3.1 FQA com Perturbações de Referência

Nesta variação, os FQAs utilizam a estratégia descrita na seção 2.1. Isso é realizado escolhendo uma função $\lambda(t)$ que descreve a perturbação de referência e aplicando-a em conjunto com β como parâmetro de U_d. Dessa forma, em cada camada, são aplicados os operadores $U_d(\lambda_k + \beta_k)U_p$, onde λ_k representa o valor da perturbação na k-ésima camada, dada por $\lambda_k = \lambda(k\Delta t)$. Assim, a evolução do estado é dada por:

$$|\psi_\ell\rangle = U_d(\lambda_\ell + \beta_\ell)U_p \ldots U_d(\lambda_k + \beta_k)U_p \ldots U_d(\lambda_2 + \beta_2)U_p\, U_d(\lambda_1 + \beta_1)U_p\, |\psi_0\rangle\,. \tag{29}$$

É importante ressaltar que λ_k deve tender a zero à medida que a quantidade de camadas se aproxima de ℓ, o total de camadas definido.

3.3.2 FQA com Função de Controle Iterativa

Nesta variação, é utilizada a estratégia descrita na seção 2.2. Inicialmente, o FQA é aplicado com ℓ camadas, resultando no conjunto de parâmetros $\vec{\beta}^{(0)} = \{\beta_k^{(0)}, k = 1, \ldots \ell\}$. Em seguida, o algoritmo passa por novas iterações. Na j-ésima iteração, o circuito é descrito pela expressão:

$$\left|\psi_\ell^{(j)}\right\rangle = U_d(\beta_\ell^{(j)})U_p \ldots U_d(\beta_2^{(j)})U_p\, U_d(\beta_1^{(j)})U_p\, |\psi_0\rangle\,. \tag{30}$$

onde $\beta_k^{(j)} = \beta_k^{(j-1)}(t) + \tilde{\beta}_k^{(j)}(t)$, com $\tilde{\beta}_k^{(j)}(t) = -A_{k-1}$.

3.3.3 FQA com Múltiplas Funções de Controle

Conforme discutido na seção 2.3, é possível utilizar mais de uma função de controle no QLC, cada uma associada a um hamiltoniano de condução diferente. O FQA pode ser facilmente adaptado para esse caso, construindo em cada camada o seguinte operador:

$$\left[\prod_j U_{d,j}(\beta_k^{(j)})\right] U_p, \tag{31}$$

onde $U_{d,j}(\beta_k^{(j)}) = e^{-i\beta_k^{(j)}\Delta t H_{d,j}}$ é o operador de evolução temporal associado ao j-ésimo hamiltoniano de condução $H_{d,j}$ e ao j-ésimo parâmetro de controle $\beta_k^{(j)}$ na k-ésima camada, que pode ser obtido por:

$$\beta_k^{(j)} = -A_{k-1}^{(j)} = -\langle\psi_{k-1}|\,[H_{d,j}, H_p]\,|\psi_{k-1}\rangle\,. \tag{32}$$

Dessa forma, o circuito resultante é dado por:

$$|\psi_\ell\rangle = \left[\prod_j U_{d,j}(\beta_\ell^{(j)})\right] U_p \cdots \left[\prod_j U_{d,j}(\beta_2^{(j)})\right] U_p \left[\prod_j U_{d,j}(\beta_1^{(j)})\right] U_p\,|\psi_0\rangle\,. \tag{33}$$

3.3.4 SO-FALQON

O SO-FALQON (*Second-Order FALQON*) é uma variação do FALQON, que utiliza uma nova lei de feedback, proposta em (30), introduzindo uma aproximação de segunda ordem em relação ao intervalo de tempo. Esta nova abordagem permite o uso de intervalos de tempo maiores, ou seja, aumenta Δt_c, acelerando a convergência do algoritmo.

Para definir a nova lei, começamos analisando a diferença da função objetivo entre duas camadas consecutivas:

$$\begin{aligned}\Delta J_k &= J_k - J_{k-1}\\ &= \langle\psi_k|\,H_p\,|\psi_k\rangle - \langle\psi_{k-1}|\,H_p\,|\psi_{k-1}\rangle\\ &= \langle\psi_{k-1}|\,U_p^\dagger U_d^\dagger(\beta_k) H_p U_d(\beta_k) U_p\,|\psi_{k-1}\rangle - \langle\psi_{k-1}|\,H_p\,|\psi_{k-1}\rangle\end{aligned} \tag{34}$$

Aplicando uma expansão de Taylor de segunda ordem em relação a Δt nos operadores de evolução temporal, onde $U_p \simeq \mathbb{1} - i\Delta t H_p - \frac{1}{2}\Delta t^2 H_p^2$ e $U_d(\beta_k) \simeq \mathbb{1} - i\beta_k \Delta t H_d - \frac{1}{2}\beta_k^2 \Delta t^2 H_d^2$, obtemos:

$$\begin{aligned}\Delta J_k \simeq\ & \Delta t \beta_k \langle\psi_{k-1}|\,i\,[H_d, H_p]\,|\psi_{k-1}\rangle\\ &+ (\Delta t)^2 (\beta_k)^2 \left\langle\psi_{k-1}\middle|\frac{1}{2}\,[[H_d, H_p], H_d]\middle|\psi_{k-1}\right\rangle\\ &+ (\Delta t)^2 \beta_k \langle\psi_{k-1}|\,[[H_d, H_p], H_p]\,|\psi_{k-1}\rangle\\ &+ \mathcal{O}\left((\Delta t)^3\right).\end{aligned} \tag{35}$$

Ao analisar a Eq. (35), observa-se que o termo de primeira ordem em Δt corresponde à derivada de J_k, indicando que a escolha de β_k faz com que a função custo diminua até a primeira ordem em relação a Δt.

Denotando $B_k = \left\langle \psi_k \left| \frac{1}{2}\left[[H_d, H_p], H_d\right] \right| \psi_k \right\rangle$ e $C_k = \langle \psi_k | [[H_d, H_p], H_p] | \psi_k \rangle$, temos:

$$\begin{aligned} \Delta J_k &\simeq \Delta t \beta_k A_{k-1} + (\Delta t)^2 (\beta_k)^2 B_{k-1} + (\Delta t)^2 \beta_k C_{k-1} + \mathcal{O}((\Delta t)^3) \\ &\simeq (\beta_k)^2 \left((\Delta t)^2 B_{k-1}\right) + \beta_k \Delta t \left(A_{k-1} + \Delta t C_{k-1}\right), \end{aligned} \tag{36}$$

em que ΔJ_k pode ser tratada como uma função quadrática em relação a β_k. Se $B_{k-1} > 0$, os valores de β_k pertencentes ao intervalo aberto $\left(0, \frac{A_{k-1}+\Delta t C_{k-1}}{\Delta t B_{k-1}}\right)$ garantem que $\Delta J_k < 0$. Em particular, o valor que proporciona a maior diminuição de J_k é dado por:

$$\beta_k = -\frac{A_{k-1} + \Delta t C_{k-1}}{2 \Delta t B_{k-1}}. \tag{37}$$

Além, para $B_{k-1} < 0$, a expressão (37) sem o sinal negativo também assegura que $\Delta J_k < 0$.Assim, a lei de feedback pode ser escrita de forma unificada como:

$$\beta_k = -\frac{A_{k-1} + \Delta t C_{k-1}}{2 \Delta t |B_{k-1}|}. \tag{38}$$

Esta é a nova lei de feedback utilizada. Em testes realizados por (30), as Eq. (38) foi aplicada quando $\left|\frac{A_{k-1}+\Delta t C_{k-1}}{2\Delta t B_{k-1}}\right| < |A_{k-1}|$; caso contrário, foi utilizada a Eq. (26).

3.3.5 WFQAE

Esta variação do FQA, proposta em (2), aprimora o método introduzido em (31), com o objetivo de estimar não apenas o estado fundamental, mas também os estados excitados de um Hamiltoniano. Esse algoritmo é chamado de Algoritmo Quântico Baseado em Feedback Ponderado para o Cálculo de Estados Excitados (*Weighted Feedback-Based Quantum Algorithm for Excited States Calculation* - WFQAE).

Para definir essa versão do algoritmo, consideremos que o espaço de Hilbert $\mathscr{H}$ tem dimensão N, ou seja, $\mathscr{H} = \mathbb{C}^N$. Neste problema, o Hamiltoniano H_p possui autovalores $E_0 < E_1 < \cdots < E_{N-1}$, com os autovetores correspondentes $|E_0\rangle, |E_1\rangle, \ldots, |E_{N-1}\rangle$, assumindo que não há degenerescência.

O algoritmo funciona considerando a dinâmica de $p+1$ sistemas quânticos:

$$\begin{aligned} i\frac{d}{dt}\left|\phi^{(0)}(t)\right\rangle &= (H_p + \beta(t) H_d)\left|\phi^{(0)}(t)\right\rangle \\ i\frac{d}{dt}\left|\phi^{(1)}(t)\right\rangle &= (H_p + \beta(t) H_d)\left|\phi^{(1)}(t)\right\rangle \\ &\vdots \\ i\frac{d}{dt}\left|\phi^{(p)}(t)\right\rangle &= (H_p + \beta(t) H_d)\left|\phi^{(p)}(t)\right\rangle \end{aligned} \tag{39}$$

com os estados iniciais $\{\left|\phi_0^{(q)}\right\rangle\}$, com $q = 0, \ldots, p$, de cada sistema ortogonais entre si, ou seja, $\left\langle \phi_0^{(q)} \mid \phi_0^{(j)} \right\rangle = \delta_{q,j}$. O objetivo é fazer com que cada estado $\left|\phi_0^{(q)}\right\rangle$ convirja para o q-ésimo menor autovetor de H_p, $|E_q\rangle$. Para isso, é utilizada a seguinte função de Lyapunov:

$$J(t) = \sum_{q=0}^{p} \omega_q \left\langle \phi^{(q)}(t) \right| H_p \left| \phi^{(q)}(t) \right\rangle, \tag{40}$$

onde o conjunto de pesos $\{\omega_q\}$ é escolhido de modo que $\omega_q > \omega_j$ para $q < j$.

A derivada temporal de $J(t)$ é dada por:

$$\begin{aligned} \frac{dJ}{dt}(t) &= \sum_{q=0}^{p} \omega_q \left\langle \phi^{(q)}(t) \right| i[H_c, H_d] \left| \phi^{(q)}(t) \right\rangle \beta(t) \\ &= \sum_{q=0}^{p} \omega_q A^{(q)}(t) \beta(t), \end{aligned} \tag{41}$$

onde $A^{(q)}(t) = \left\langle \phi^{(q)}(t) \right| i[H_c, H_d] \left| \phi^{(q)}(t) \right\rangle$. Este termo será negativo se $\beta(t)$ tiver a forma:

$$\beta(t) = -\omega f \left(\sum_{q=0}^{p} \omega_q A^{(q)}(t), t \right), \tag{42}$$

com as mesmas propriedades da função apresentada na Eq. (6). Considerando o caso mais simples para $\beta(t)$, temos:

$$\beta(t) = -\omega \sum_{q=0}^{p} \omega_q A^{(q)}(t). \tag{43}$$

A função $J(t)$ é minimizada quando todos os sistemas evoluem para o estado fundamental. No entanto, devido à ortogonalidade inicial e ao uso de uma única função de controle, os sistemas convergem para diferentes estados. Assim, os pesos ω_q orientam a evolução para que os estados de menor energia sejam associados aos maiores pesos, favorecendo a ocupação dos autovalores mais baixos por índices menores de q.

O WFQAE funciona da seguinte forma. Inicialmente, são escolhidos o passo de tempo Δt e a quantidade de camadas ℓ. Em seguida, são preparados $p+1$ circuitos quânticos, onde o q-ésimo circuito é inicializado com um estado $\left|\phi_0^{(q)}\right\rangle$, que deve ser fácil de preparar e ortogonal aos estados dos demais circuitos. A construção das camadas é realizada em cada iteração, onde em cada camada k são aplicados os operadores $U_d(\beta_k)U_p$ em cada circuito, obtendo-se o estado $\left|\phi_k^{(q)}\right\rangle$:

$$
\begin{aligned}
\left|\phi_k^{(0)}\right\rangle &= U_d(\beta_k)U_p \ldots U_d(\beta_2)U_p\, U_d(\beta_1)U_p \left|\phi_0^{(0)}\right\rangle \\
\left|\phi_k^{(1)}\right\rangle &= U_d(\beta_k)U_p \ldots U_d(\beta_2)U_p\, U_d(\beta_1)U_p \left|\phi_0^{(1)}\right\rangle \\
&\vdots \\
\left|\phi_k^{(p)}\right\rangle &= U_d(\beta_k)U_p \ldots U_d(\beta_2)U_p\, U_d(\beta_1)U_p \left|\phi_0^{(p)}\right\rangle
\end{aligned}
\tag{44}
$$

Em seguida, é feita a medida de $A_k^{(q)} = \left\langle \phi_k^{(q)} \right| i[H_c, H_d] \left| \phi_k^{(q)} \right\rangle$ em cada circuito. Para a construção da próxima camada, β_{k+1} é obtido pela lei de feedback:

$$
\beta_{k+1} = -\omega \sum_{q=0}^{p} \omega_q A_k^{(q)}. \tag{45}
$$

Esse processo é repetido até a camada ℓ. A figura 3 mostra o diagrama do WFQAE.

Se ℓ for a quantidade suficiente de camadas, obtém-se:

$$
\begin{aligned}
\left|\phi_\ell^{(0)}\right\rangle &\approx |E_0\rangle \\
\left|\phi_\ell^{(1)}\right\rangle &\approx |E_1\rangle \\
&\vdots \\
\left|\phi_\ell^{(p)}\right\rangle &\approx |E_p\rangle\,.
\end{aligned}
\tag{46}
$$

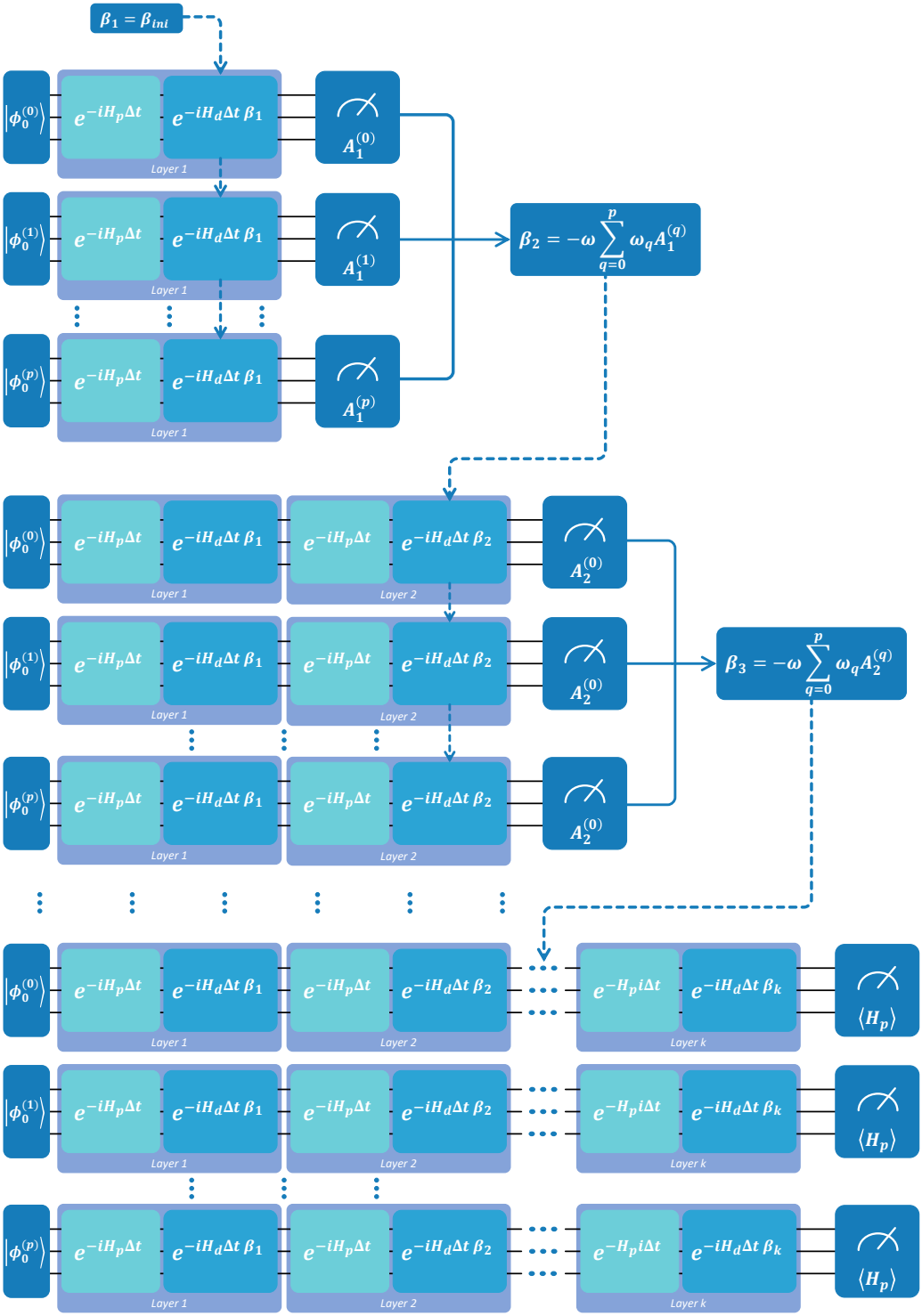

Figura 3 – Diagrama ilustrativo de um WFQAE. O algoritmo começa com a escolha do passo de tempo Δt e do número de camadas ℓ. Em seguida, $p+1$ circuitos quânticos são preparados, cada um inicializado em um estado $\left|\phi_0^{(q)}\right\rangle$, que é fácil de preparar e ortogonal aos estados dos outros circuitos. A construção das camadas ocorre iterativamente, com a aplicação dos operadores $U_d(\beta_k)U_p$ em cada circuito, resultando nos estados $\left|\phi_k^{(q)}\right\rangle$. A configuração final minimiza a função de Lyapunov $J(t)$ associando os estados de menores valores esperados aos maiores pesos, otimizando a convergência para os autovalores do Hamiltoniano. Figura adaptada de (2).

4 Codificação de Problemas de Otimização para Algoritmos Quânticos

Neste capítulo, vamos explicar como problemas de otimização combinatória, em especial aqueles do tipo QUBO, podem ser codificados para serem resolvidos por algoritmos quânticos.

4.1 Otimização Binária Quadrática Irrestrita

A Otimização Binária Quadrática Irrestrita (*Quadratic Unconstrained Binary Optimization* - QUBO) (32, 33, 34) é um problema de otimização combinatória que envolve a minimização de uma função quadrática com restrições binárias nas variáveis, as quais assumem valores de 0 ou 1. Esse tipo de problema é relevante em diversas áreas, como logística (35, 36, 37), engenharia (38) e inteligência artificial (39, 40).

A forma geral da função objetivo de um problema QUBO é dada por:

$$C(x) = \sum_{i,j} x_i Q_{ij} x_j + \sum_i c_i x_i = x^T Q x + c^T x \tag{47}$$

onde $x = (x_1, x_2, \ldots, x_n)$ é o vetor de variáveis binárias ($x_i \in \{0, 1\}$) de dimensão n, Q é uma matriz simétrica que contém coeficientes quadráticos $Q_{i,j}$, e c representa uma matriz coluna que armazena os coeficientes lineares c_i. O objetivo é encontrar a configuração de x que minimize $C(x)$.

Um ponto importante é que o número total de configurações possíveis cresce exponencialmente com o número de variáveis: como cada x_i pode assumir dois valores, o espaço de busca tem dimensão 2^n. Por exemplo, para $n = 10$, há $2^{10} = 1.024$ possíveis soluções; para $n = 20$, esse número já ultrapassa um milhão. Para valores mais altos de n, como $n = 50$, o total de configurações possíveis chega à ordem de 10^{15}, tornando inviável a busca exaustiva em arquiteturas clássicas.

Esse aumento rápido na dimensão do espaço de busca é uma das principais dificuldades na resolução de problemas QUBO de grande escala, justificando o uso de algoritmos quânticos e heurísticas inspiradas em física para explorar o espaço de busca de forma mais eficiente.

4.1.1 Codificação de Restrições

Embora o QUBO seja, por definição, um modelo irrestrito, é possível incorporar restrições à formulação por meio de termos de penalização adicionados à função objetivo (41). A ideia é transformar o problema com restrições em um problema sem restrições, em que soluções que

violam as condições impostas recebem penalidades proporcionais ao grau de violação, o que resulta no aumento de seu custo.

Consideremos uma restrição de igualdade da forma:

$$\sum_i a_i x_i = b. \tag{48}$$

Essa condição pode ser incorporada ao QUBO adicionando à função objetivo o seguinte termo de penalização quadrática:

$$P_{\mathsf{eq}}(x) = \left(\sum_i a_i x_i - b\right)^2, \tag{49}$$

o qual assume valor zero quando a igualdade é satisfeita, e aumenta conforme a soma se afasta de b.

A nova função de custo total é então escrita como:

$$C'(x) = C(x) + \lambda P_{\mathsf{eq}}(x), \tag{50}$$

onde $\lambda > 0$ é um parâmetro que determina o peso da penalidade. Valores altos de λ aumentam a penalização das soluções que violam a restrição, incentivando o algoritmo a evitá-las.

Esse procedimento pode ser estendido para múltiplas restrições de igualdade, bastando somar os respectivos termos penalizadores.

Restrições de desigualdade, do tipo:

$$\sum_i a_i x_i \leq W \quad \text{ou} \quad \sum_i a_i x_i \geq W,$$

também podem ser tratadas, embora exijam estratégias adicionais, pois a penalidade só deve ser aplicada quando a desigualdade for violada. Uma abordagem comum consiste em introduzir variáveis auxiliares binárias e reescrever a desigualdade como uma igualdade sujeita a penalização.

Considere a desigualdade:

$$\mathcal{W} = \sum_i a_i x_i \leq W, \quad \text{com } a_i > 0.$$

Essa condição pode ser codificada de duas formas, ambas envolvendo variáveis auxiliares.

Na primeira abordagem, emprega-se uma codificação *one-hot* para as variáveis auxiliares, introduzindo $y_j \in \{0, 1\}$ para $j = 0, \ldots, W$, onde $y_j = 1$ indica que o valor de $\mathcal{W} = \sum_i a_i x_i$ é igual a j. Para que essa representação seja válida, é necessário impor duas condições: que

apenas uma das variáveis y_j assuma o valor 1 e que a combinação ponderada $\sum_j jy_j$ coincida com o valor de $\sum_i a_i x_i$. Ambas as condições podem ser implementadas por meio do seguinte termo de penalização:

$$P_{\text{in}}(x,y) = \left(1 - \sum_{j=0}^{W} y_j\right)^2 + \left(\sum_i a_i x_i - \sum_{j=0}^{W} jy_j\right)^2. \tag{51}$$

Essa formulação assegura que $\mathcal{W} \in [0, W]$, satisfazendo a desigualdade. No entanto, exige o uso de $W+1$ variáveis auxiliares, o que pode ser inviável para valores grandes de W.

No segundo método, a desigualdade é reescrita como uma igualdade da forma:

$$\sum_i a_i x_i + Y = W, \tag{52}$$

em que Y representa a diferença entre W e $\sum_i a_i x_i$, podendo assumir valores inteiros entre 0 e W. Para modelar Y, introduz-se um conjunto de variáveis auxiliares binárias $y_j \in \{0,1\}$, que formam sua representação binária:

$$Y = \sum_{j=0}^{\lceil \log_2(W+1) \rceil - 1} 2^j y_j. \tag{53}$$

Com essa reformulação, a desigualdade $\sum_i a_i x_i \leq W$ é transformada em uma igualdade penalizada:

$$P_{\text{in}}(x,y) = \left(W - \sum_i a_i x_i - Y\right)^2. \tag{54}$$

Essa abordagem exige apenas $\lceil \log_2(W+1) \rceil$ variáveis auxiliares, o que a torna mais eficiente em termos de recursos para valores elevados de W.

Nas duas formas, a nova função custo pode ser escrita como:

$$C'(x,y) = C(x) + \lambda P_{in}(x,y) \tag{55}$$

onde as variáveis auxiliares $y = (y_0, y_1, \dots)$ são incorporados ao conjunto de variáveis do problema QUBO.

4.2 Transcrição de um Problema QUBO para um Hamiltoniano

Uma abordagem para resolver um problema QUBO com algoritmos quânticos consiste em transcrevê-lo para um Hamiltoniano quântico (42). Nesse mapeamento, cada variável binária $x_i \in \{0,1\}$ é associada ao estado de um qubit $|x_i\rangle$, onde os valores 0 e 1 correspondem, respectivamente, aos estados da base computacional $|0\rangle$ e $|1\rangle$.

O objetivo é construir um Hamiltoniano H_f tal que:

$$H_f|x\rangle = C(x)|x\rangle,$$

onde cada estado $|x\rangle = |x_{L-1}\rangle \otimes \cdots \otimes |x_1\rangle \otimes |x_0\rangle$ representa um elemento da base computacional em um espaço de Hilbert de dimensão 2^L. Assim, o valor da função objetivo $C(x)$, associada ao problema QUBO, torna-se o autovalor correspondente ao estado $|x\rangle$.

Para realizar esse mapeamento, utiliza-se a relação entre os autovalores do operador de Pauli σ_i^z e os estados computacionais:

$$\begin{cases} \sigma_i^z|0\rangle = +|0\rangle, \\ \sigma_i^z|1\rangle = -|1\rangle, \end{cases}$$

o que implica que:

$$\sigma_i^z|x_i\rangle = (1 - 2x_i)|x_i\rangle.$$

Com isso, define-se o operador:

$$\tilde{z}_i = \frac{\mathbb{1} - \sigma_i^z}{2},$$

o qual satisfaz $\tilde{z}_i|x_i\rangle = x_i|x_i\rangle$, permitindo encontrar o Hamiltoniano ao substituir diretamente as variáveis x_i da função custo $C(x)$ por operadores quânticos.

Dessa forma, o Hamiltoniano que implementa o problema QUBO é dado por:

$$H_f = \sum_{i,j} Q_{ij}\tilde{z}_i\tilde{z}_j + \sum_i c_i\tilde{z}_i,$$

onde os coeficientes Q_{ij} e c_i provêm da formulação original do QUBO.

O estado fundamental desse Hamiltoniano corresponde à solução ótima do problema. Esse estado pode ser obtido por meio de algoritmos de otimização quântica, minimizando-se o valor esperado da energia:

$$\min_{|\psi\rangle} \langle\psi| H_f |\psi\rangle .$$

Caso existam múltiplas soluções ótimas (ou seja, configurações x distintas que minimizam $C(x)$), o estado fundamental será degenerado, e o subespaço de menor energia será formado por todos os vetores $|x\rangle$ associados a essas soluções.

4.3 MaxCut

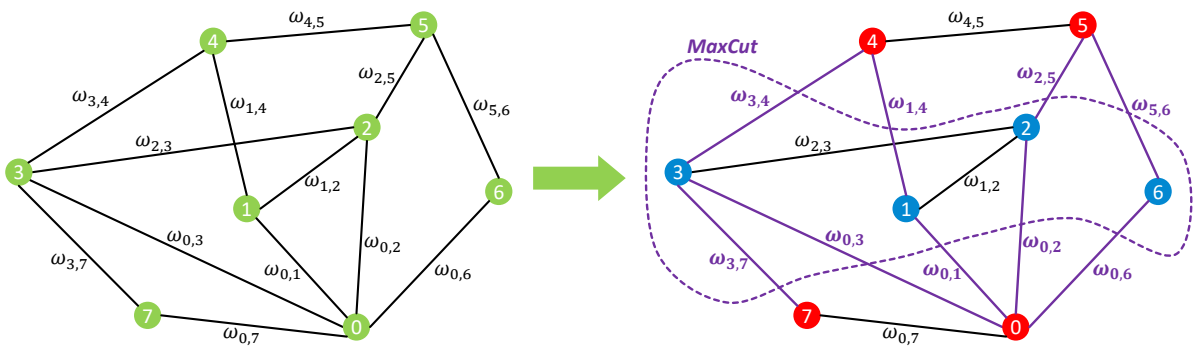

Figura 4 – Ilustração da partição de um grafo no problema MaxCut, onde as arestas entre vértices de subconjuntos distintos compõem o corte. Figura elaborada pelo autor.

O problema MaxCut (22, 24) é um problema de otimização combinatória em grafos. Dado um grafo $G = (V, E)$, o objetivo é particionar o conjunto de vértices V em dois subconjuntos disjuntos S e S', de modo a maximizar a soma dos pesos das arestas $e \in E$ que conectam vértices de subconjuntos diferentes. Essas arestas formam o chamado corte (*cut*) do grafo.

A formulação QUBO do MaxCut consiste em associar a cada vértice $i \in V$ uma variável binária $x_i \in \{0, 1\}$, onde os valores 0 e 1 indicam a qual subconjunto o vértice pertence. Considerando um grafo com L vértices numerados de 0 a $L-1$, a função custo a ser minimizada é:

$$C(x) = -\sum_{i,j=0}^{L-1} w_{ij} x_i (1 - x_j), \tag{56}$$

onde w_{ij} representa o peso da aresta entre os vértices i e j. O termo $x_i(1 - x_j) + x_j(1 - x_i)$ é igual a 1 quando $x_i \neq x_j$ (ou seja, quando a aresta (i, j) é cortada) e 0 quando $x_i = x_j$, indicando que os vértices pertencem ao mesmo subconjunto. Logo, essa formulação busca maximizar o corte, isto é, a soma dos pesos das arestas que ligam vértices em subconjuntos diferentes. No entanto, como a função custo está precedida por um sinal negativo, o problema é reescrito como um problema de minimização.

Reescrevendo a função custo:

$$
\begin{aligned}
C(x) &= -\sum_{\substack{i=0\\ j>i}}^{L-1} w_{ij}(x_i(1-x_j)+x_j(1-x_i)) \\
&= -\sum_{\substack{i=0\\ j>i}}^{L-1} w_{ij}(x_i+x_j-2x_ix_j),
\end{aligned}
\tag{57}
$$

e substituindo x_i pelos operadores $\tilde{z}_i$, obtemos o Hamiltoniano associado ao problema MaxCut:

$$
\begin{aligned}
H &= -\sum_{\substack{i=0\\ j>i}}^{L-1} w_{ij}(\tilde{z}_i+\tilde{z}_j-2\tilde{z}_i\tilde{z}_j) \\
&= -\sum_{\substack{i=0\\ j>i}}^{L-1} \frac{w_{ij}}{2}(\mathbb{1}-\sigma_i^z\sigma_j^z)
\end{aligned}
\tag{58}
$$

A Eq. (58) é a formulação mais comum do Hamiltoniano para o MaxCut. No entanto, é possível simplificá-la ainda mais:

$$
H = -\sum_{\substack{i=0\\ j>i}}^{L-1} \frac{w_{ij}}{2}\mathbb{1} + \sum_{\substack{i=0\\ j>i}}^{L-1} \frac{w_{ij}}{2}\sigma_i^z\sigma_j^z \tag{59}
$$

O primeiro termo apenas desloca os níveis de energia e não afeta os autoestados do Hamiltoniano, podendo ser descartado. A constante multiplicativa $1/2$ no segundo termo também não altera os autoestados, podendo ser omitida. Assim, o Hamiltoniano assume a forma:

$$
H = \sum_{\substack{i=0\\ j>i}}^{L-1} w_{ij}\sigma_i^z\sigma_j^z \tag{60}
$$

5 Simetrias

Simetrias são transformações invertíveis que, quando aplicadas a um sistema físico, não alteram suas propriedades observáveis, ou seja, o sistema permanece invariante sob essa transformação (43). Em mecânica quântica, tais transformações são descritas por operadores unitários ou antiunitários (44). Neste contexto, será considerado apenas o caso de transformações unitárias que deixam o Hamiltoniano do sistema H invariante. Assim, dada uma transformação P,

$$P^\dagger H P = H. \tag{61}$$

Como P é unitário ($P^\dagger = P^{-1}$), essa condição é equivalente a afirmar que P comuta com o Hamiltoniano, isto é, $[P, H] = 0$.

De forma geral, a presença de simetrias em um Hamiltoniano favorece a degeneração dos autovalores. Em particular, quando existem simetrias que não comutam entre si, o Hamiltoniano terá necessariamente pelo menos um autovalor degenerado. Nesses casos, a energia isoladamente não é suficiente para caracterizar completamente os autoestados, já que múltiplos estados podem corresponder ao mesmo valor de energia.

Nesse contexto, as simetrias são relevantes porque permitem identificar conjuntos de operadores que comutam entre si e com o Hamiltoniano. Isso facilita a diagonalização de H e possibilita uma classificação mais precisa dos autoestados, com base nos autovalores dos operadores associados às simetrias do sistema.

Seja $|p\rangle$ um autoestado de um operador de simetria P, com autovalor ϕ_p. Pela comutação entre P e H, temos que:

$$P(H\,|p\rangle) = HP\,|p\rangle = \phi_p(H\,|p\rangle), \tag{62}$$

ou seja, $H\,|p\rangle$ também é um autoestado de P com o mesmo autovalor ϕ_p.

Se ϕ_p for não degenerado, então $H|p\rangle$ será automaticamente proporcional a $|p\rangle$, ou seja, $|p\rangle$ será simultaneamente um autovetor de P e de H. No caso de degenerescência, $H|p\rangle$ permanece no subespaço associado ao autovalor ϕ_p, que chamamos de setor $\mathcal{E}_p$. O número quântico p serve como a etiqueta que identifica o setor $\mathcal{E}_p$ ao qual o estado pertence. Dessa forma, $\mathcal{E}_p$ é invariante sob a ação de H.

Além disso, setores diferentes $\mathcal{E}_p$ e $\mathcal{E}_{p'}$ com $p \neq p'$ são ortogonais entre si:

$$\langle p|p'\rangle = 0, \quad \text{com} \quad |p\rangle \in \mathcal{E}_p, \quad |p'\rangle \in \mathcal{E}_{p'}. \tag{63}$$

Essa ortogonalidade assegura que a ação do Hamiltoniano sobre um subespaço não afeta os outros, permitindo que H seja tratado de maneira independente em cada subespaço. Assim, a solução de H pode ser dividida em problemas menores, restritos a cada subespaço.

Representando H na base dos autovetores de P, $\{|r,p\rangle\}$, onde r é um índice que distingue os estados degenerados com o mesmo autovalor ϕ_p, a matriz do Hamiltoniano satisfaz:

$$\langle r,p|H|r',p'\rangle = 0 \quad \text{para} \quad p \neq p', \tag{64}$$

o que implica que H assume uma forma bloco-diagonal:

$$\mathrm{H} = \mathrm{H}_0 \oplus \mathrm{H}_1 \oplus \cdots \oplus \mathrm{H}_p \oplus \cdots\ , \tag{65}$$

onde cada H_p é a matriz do Hamiltoniano restrita ao subespaço $\mathcal{E}_p$. Cada bloco pode ser diagonalizado separadamente, simplificando a análise do sistema.

Se o sistema possuir mais de um operador de simetria, $\{P_1, P_2, \ldots, P_n\}$, todos comutando entre si, a estrutura de blocos pode ser refinada ainda mais. Nesse caso, os estados podem ser simultaneamente classificados pelos autovalores $(\phi_{p_1}, \phi_{p_2}, \ldots, \phi_{p_n})$ associados a cada simetria. A base usada será formada por autovetores comuns a esses operadores,

$$|r, p_1, p_2, \ldots, p_n\rangle\ , \tag{66}$$

e o Hamiltoniano terá uma decomposição em blocos menores, cada um associado a um conjunto especifico de números quânticos.

Se, em cada setor, não houver degenerescência adicional, ou seja, se para cada conjunto de números quânticos houver no máximo um autoestado de H associado a um mesmo autovalor de energia, então todos os autoestados de H podem ser identificados pelos números quânticos do setor junto com a energia. Nesse caso, o conjunto formado pelos operadores de simetria e pelo Hamiltoniano constitui um conjunto completo de operadores compatíveis.

5.1 Conservação dos números quânticos na evolução temporal

A invariância de um operador de simetria P sob a ação de um Hamiltoniano dependente do tempo $H(t)$ implica na conservação do número quântico associado ao longo da evolução temporal do sistema. Suponha que, no instante inicial, o sistema esteja no autoestado $|p\rangle$ de P. A evolução temporal é descrita pelo operador $U(t)$. Se em todos os instantes $H(t)$ comuta com P, então $U(t)$ também comuta com P. Assim:

$$P|p(t)\rangle = PU(t)|p\rangle = U(t)P|p\rangle = \phi_p U(t)|p\rangle = \phi_p|p(t)\rangle, \tag{67}$$

mostrando que o estado evoluído permanece no setor de P, com o número quântico p conservado. Isso significa que a dinâmica não mistura setores diferentes, e o vetor continua pertencendo ao mesmo subespaço $\mathcal{E}_p$.

5.2 Simetrias Cíclicas

Considere que P é um operador de simetria cíclica de ordem n ($\mathbb{Z}_n$) (45), ou seja, satisfaz:

$$P^n = \mathbb{1}. \tag{68}$$

Como P é um operador unitário, seus autovalores ϕ devem ter módulo 1 e podem ser escritos como $\phi_p = e^{i\delta_p}$, com δ_p real (46). Vamos buscar os autovalores e autovetores de P. Seja $|p\rangle$ um autovetor de P, então $P|p\rangle = \phi_p |p\rangle$.

Aplicando P n vezes:

$$P^n |p\rangle = (\phi_p)^n |p\rangle = e^{in\delta_p} |p\rangle . \tag{69}$$

Mas como $P^n = \mathbb{1}$, temos que $e^{in\delta_p} = 1$. Portanto, $n\delta_p$ deve ser um múltiplo inteiro de 2π:

$$n\delta_p = 2\pi p, \quad p \in \mathbb{Z}, \tag{70}$$

o que implica:

$$\delta_p = \frac{2\pi p}{n}. \tag{71}$$

Assim, os autovalores de P são discretos e dados por:

$$\phi_p = e^{i\frac{2\pi p}{n}}, \tag{72}$$

onde $p = 0, 1, \ldots, n-1$.

Uma vez determinados os autovalores $\phi_p = e^{i\frac{2\pi p}{n}}$, a construção explícita dos autovetores de P pode ser realizada da seguinte forma.

Considere um conjunto de estados $\{|\psi_j\rangle\}$ que formam uma base na qual P atua de maneira cíclica:

$$P|\psi_j\rangle = |\psi_{j+1}\rangle \quad (\text{mod } n), \tag{73}$$

ou seja, a ação de P apenas permuta ciclicamente os estados da base. Definindo uma relação de equivalência na qual dois estados são considerados equivalentes se um pode ser obtido do outro por aplicações sucessivas de P, o conjunto $\{|\psi_j\rangle\}$ forma uma classe de equivalência sob a ação de P (47).

Buscamos autovetores $|p\rangle$ de P associados ao autovalor ϕ_p, que podem ser escritos como uma combinação linear dos estados $\{|\psi_j\rangle\}$:

$$|p\rangle = \sum_{j=0}^{n-1} c_j \, |\psi_j\rangle \, . \tag{74}$$

Aplicando P a $|p\rangle$, temos:

$$P\,|p\rangle = \sum_{j=0}^{n-1} c_j P\,|\psi_j\rangle = \sum_{j=0}^{n-1} c_j \, |\psi_{j+1}\rangle \, . \tag{75}$$

Por outro lado, como $|p\rangle$ é autovetor de P com autovalor ϕ_p, também:

$$P\,|p\rangle = e^{i\delta_p}\,|p\rangle = e^{i\delta_p} \sum_{j=0}^{n-1} c_j \, |\psi_j\rangle \, . \tag{76}$$

Comparando as equações (75) e (76), obtemos a seguinte relação entre os coeficientes:

$$c_{j-1} = c_j e^{i\delta_p}, \tag{77}$$

o que implica que:

$$c_j = c_{j-1} e^{-i\delta_p} = c_{j-2} e^{-2i\delta_p} = \cdots = c_0 e^{-i\delta_p j}. \tag{78}$$

Portanto, todos os coeficientes c_j são determinados a partir de c_0. Para fixar a normalização do vetor $|p\rangle$, escolhemos $c_0 = \mathbb{C}$ como constante de normalização, resultando na forma final:

$$|p\rangle = \mathbb{C} \sum_{j=0}^{n-1} e^{-i\delta_p j} \, |\psi_j\rangle \, . \tag{79}$$

Substituindo agora a ação de P na base cíclica, que permite escrever $|\psi_j\rangle$ como:

$$|\psi_j\rangle = P^j \, |\psi_0\rangle \, , \tag{80}$$

a expressão do autovetor $|p\rangle$ pode ser reescrita como:

$$|p\rangle = \mathbb{C} \sum_{j=0}^{n-1} \exp\left(-i\frac{2\pi p}{n} j\right) P^j \, |\psi_0\rangle \, . \tag{81}$$

Essa expressão mostra que o vetor $|p\rangle$ é obtido pela aplicação de um operador linear ao estado $|\psi_0\rangle$. Tal operador é:

$$\Pi^{(p)} = \frac{1}{n}\sum_{j=0}^{n-1} \exp\left(-i\frac{2\pi p}{n}j\right) P^j, \tag{82}$$

o qual satisfaz $\left(\Pi^{(p)}\right)^2 = \Pi^{(p)}$ e, portanto, é um projetor sobre o subespaço associado ao autovalor $e^{2\pi i p/n}$. Assim, a construção do autovetor $|p\rangle$ pode ser interpretada como a projeção do estado $|\psi_0\rangle$ nesse subespaço, seguida de uma normalização:

$$|p\rangle = \frac{\Pi^{(p)}|\psi_0\rangle}{\sqrt{\langle\psi_0|\,\Pi^{(p)}|\psi_0\rangle}}. \tag{83}$$

No entanto, pode ocorrer que $\Pi^{(p)}|\psi_0\rangle = 0$, caso $|\psi_0\rangle$ não tenha componente nas direções do autoespaço associado a $e^{2\pi i p/n}$. Isso está relacionado à linearidade dos vetores $\{|\psi_0\rangle, P|\psi_0\rangle, \ldots, P^{n-1}|\psi_0\rangle\}$. Se forem linearmente independentes, todos os autovalores terão autovetores não nulos gerados por $\Pi^{(p)}$. Caso contrário, será necessário escolher outros vetores iniciais para completar a base de autovetores de P (48).

6 Modelo ANNNI

O modelo de Ising Quântico Anisotrópico com Interações de Segundos Vizinhos, conhecido como modelo ANNNI (*Axial Next-Nearest Neighbor Ising*) (49, 3), é uma extensão do modelo de Ising tradicional. Ele descreve o comportamento de spins-$1/2$ organizados em uma cadeia unidimensional, onde os spins interagem anisotropicamente tanto com seus primeiros vizinhos, quanto com os segundos vizinhos.

O Hamiltoniano que descreve o modelo ANNNI é dado por:

$$H_A(\kappa, g) = -J \sum_{j=0}^{L-1} \left(\sigma_j^z \sigma_{j+1}^z - \kappa \sigma_j^z \sigma_{j+2}^z + g \sigma_j^x \right) \tag{84}$$

onde L é o tamanho da cadeia de spins e σ_j^x, σ_j^y e σ_j^z são as matrizes de Pauli que correspondem aos componentes do spin no sítio j. Condições periódicas de contorno são aplicadas, ou seja, $\sigma_L{=}\sigma_0$. O parâmetro J é a constante de acoplamento para a interação ferromagnética de primeiro vizinho, que é positiva e define a escala de energia ($J = 1$). O termo κ é a constante de acoplamento adimensional que descreve a intensidade da interação antiferromagnética entre os segundos vizinhos. Já o termo g é um parâmetro que descreve a intensidade do campo magnético transversal que age no sistema.

O diagrama de fases do estado fundamental do modelo ANNNI, representado na

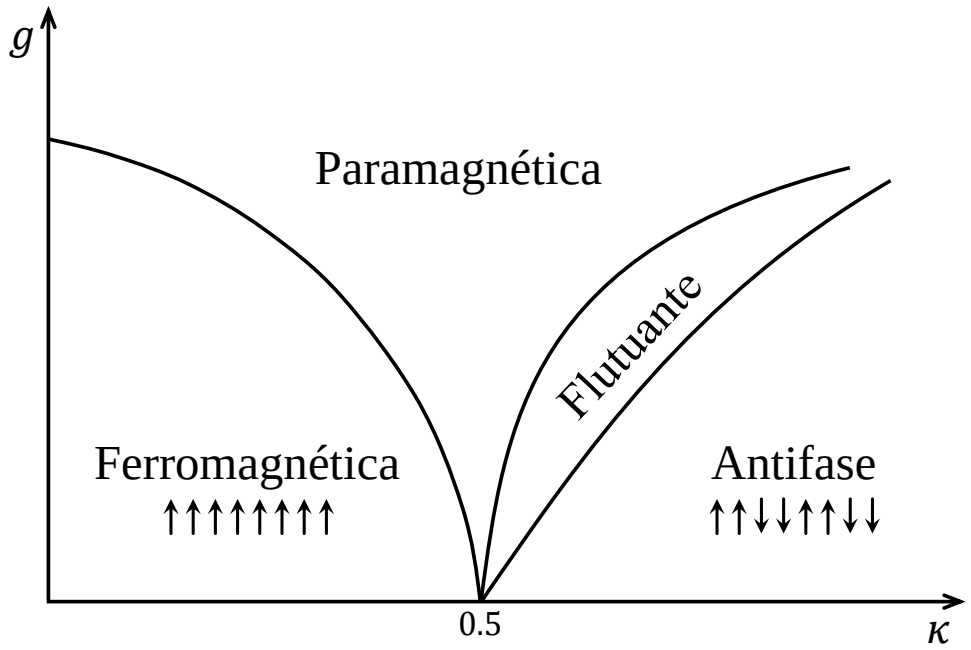

Figura 5 – Representação esquemática do diagrama de fases do modelo ANNNI sob campo transverso. Adaptado de (3, 4, 5).

Figura 5, que inclui fases ferromagnética, antifase, paramagnética e flutuante (50, 51, 52), é resultado da frustração entre as interações ferromagnéticas de primeiro vizinho e as interações antiferromagnéticas de segundo vizinho, bem como da tendência de desordem introduzida pelo campo transversal.

A fase ferromagnética ocorre quando a interação de primeiro vizinho é predominante, com os estados mais prováveis sendo aqueles em que os spins estão alinhados em uma única direção ($\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow$). Quando a interação de segundo vizinho prevalece, uma nova solução de compromisso leva à antifase com uma periodicidade de 4 sítios ($\uparrow\uparrow\downarrow\downarrow\uparrow\uparrow\downarrow\downarrow$). Quando o campo transversal é predominante, os spins tendem a se alinharem com o campo, criando a fase paramagnética. Na fase flutuante (53, 54), localizada entre a fase paramagnética e a antifase, as correlações entre os spins decaem de forma algébrica, indicando uma ordem de longo alcance modulada que varia com o parâmetro de frustração κ, sem uma ordem de longo alcance fixa.

6.1 Simetrias do Modelo ANNNI

No Hamiltoniano do modelo ANNNI, não há termos que favoreçam uma orientação particular dos spins ao longo do eixo z. Cada spin interage apenas com seus primeiros e segundos vizinhos, e a intensidade dessas interações é independente da posição na cadeia. Assim, a energia do sistema depende apenas das orientações relativas entre spins vizinhos, e não de sua localização específica. Como resultado, o modelo possui três simetrias principais: inversão de spin, reflexão espacial e translação (incluindo suas potências). Essas transformações preservam as interações entre vizinhos e mantêm o Hamiltoniano invariante.

A inversão de spin é representada pelo operador I, que inverte o sinal dos spins em cada sítio, de modo que $I^\dagger \sigma_i^z I = -\sigma_i^z$. Sua forma é:

$$I = \prod_{i=0}^{L-1} \sigma_i^x \tag{85}$$

Um exemplo da ação de I é:

$$|\uparrow\uparrow\downarrow\downarrow\uparrow\uparrow\rangle \to |\downarrow\downarrow\uparrow\uparrow\downarrow\downarrow\rangle \,.$$

O operador I é cíclico de ordem dois, e seus autovalores são $\phi_{p_I} = \pm 1$. O projetor correspondente ao setor p_I é dado por:

$$\Pi_I^{(p_I)} = \frac{1}{2}(\mathbb{1} + \phi_{p_I} I), \tag{86}$$

de modo que os autovetores podem ser escritos como:

$$|j, p_I\rangle = \mathbb{C}\, \Pi_I^{(p_I)} \, |\psi\rangle \,, \tag{87}$$

onde j é um índice de degenerescência.

A reflexão espacial é representada pelo operador R, que inverte a ordem dos spins em relação ao centro da cadeia, com $R^\dagger \sigma_i^z R = \sigma_{L-1-i}^z$. Sua expressão é:

$$R = \prod_{i=0}^{\lfloor (L-1)/2 \rfloor} \left(\frac{\mathbb{1} + \sigma_i^x \sigma_{L-1-i}^x + \sigma_i^y \sigma_{L-1-i}^y + \sigma_i^z \sigma_{L-1-i}^z}{2} \right), \tag{88}$$

Sua ação sobre um estado é:

$$R \left| d_0 \, d_1 \, \cdots \, d_{L-2} \, d_{L-1} \right\rangle = \left| d_{L-1} \, d_{L-2} \, \cdots \, d_1 \, d_0 \right\rangle .$$

Assim como I, R é de ordem dois, com autovalores $\phi_{p_R} = \pm 1$. O projetor correspondente é:

$$\Pi_R^{(p_R)} = \frac{1}{2}(\mathbb{1} + \phi_{p_R} R), \tag{89}$$

com autovetores definidos como:

$$\left| j, p_R \right\rangle = \mathbb{C} \, \Pi_R^{(p_R)} \left| \psi \right\rangle . \tag{90}$$

Autovetores com $\phi_{p_R} = +1$ são ditos pares, enquanto aqueles com $\phi_{p_R} = -1$ são ímpares.

A translação é representada pelo operador T, que desloca ciclicamente os sítios da cadeia: $T^\dagger \sigma_i^z T = \sigma_{i+1}^z$, com forma:

$$T = \prod_{i=0}^{L-2} \left(\frac{\mathbb{1} + \sigma_i^x \sigma_{i+1}^x + \sigma_i^y \sigma_{i+1}^y + \sigma_i^z \sigma_{i+1}^z}{2} \right) \tag{91}$$

Sua ação é:

$$T \left| d_0 \, d_1 \, \cdots \, d_{L-2} \, d_{L-1} \right\rangle = \left| d_1 \, d_2 \, \cdots \, d_{L-1} \, d_0 \right\rangle .$$

O operador T é cíclico de ordem L e possui autovalores dados por $\phi_{p_T} = e^{i \frac{2\pi}{L} p_T}$, onde $p_T = 0, 1, \ldots, L-1$. O índice p_T é chamado de número de onda e usualmente é denotado por k. O projetor para o setor p_T é:

$$\Pi_T^{(p_T)} = \frac{1}{L} \sum_{l=0}^{L-1} \left(\phi_{p_T}^{-1} T \right)^l , \tag{92}$$

de modo que os autovetores associados são:

$$\left| j, p_T \right\rangle = \mathbb{C} \, \Pi_T^{(p_T)} \left| \psi \right\rangle . \tag{93}$$

Embora os operadores I, R e T comutem com o Hamiltoniano do modelo ANNNI, R e T não comutam entre si em geral, já que $R^\dagger T R = T^{-1}$. Nesses casos, utiliza-se I e T como conjunto de operadores compatíveis.

No entanto, para L par, nos setores com $p_T = 0$ ou $p_T = L/2$, os autovalores de T são $\phi_{p_T} = \pm 1$, o que implica $T^2 = \mathbb{1}$, e portanto $T = T^{-1}$. Isso garante que R e T comutam nesses subespaços. Nessa situação, é possível construir uma base comum de autovetores para R e T, permitindo o uso simultâneo de p_R e $p_T \in \{0, L/2\}$ como números quânticos adicionais.

Assim, os vetores pertencentes aos setores podem ser rotulados pelos números quânticos p_I, p_T e p_R:

$$|j, p_I, p_T, p_R\rangle \, .$$

onde, nos casos em que o setor não pode ser caracterizado por p_R, utiliza-se o símbolo "$-$".

A partir da base produto dos autoestados de σ_z, $|s\rangle \in \{|\uparrow\rangle, |\downarrow\rangle\}^{\otimes L}$, é possível projetar os vetores dessa base no setor definido pelos valores de p_I e p_T:

$$|j, p_I, p_T\rangle = \frac{\Pi_T^{(p_T)} \Pi_I^{(p_I)} |s\rangle}{\sqrt{\langle s| \Pi_T^{(p_T)} \Pi_I^{(p_I)} |s\rangle}}. \tag{94}$$

formando uma base para esse setor.

Nos setores onde $p_T = 0$ ou $p_T = L/2$, aplica-se adicionalmente o projetor $P_R^{(p_R)}$, resultando em:

$$|j, p_I, p_T, p_R\rangle = \frac{\Pi_R^{(p_R)} \Pi_T^{(p_T)} \Pi_I^{(p_I)} |s\rangle}{\sqrt{\langle s| \Pi_R^{(p_R)} \Pi_T^{(p_T)} \Pi_I^{(p_I)} |s\rangle}}. \tag{95}$$

Com essa abordagem, é possível construir uma base para cada combinação admissível dos números quânticos p_I, p_T e, quando aplicável, p_R. Nessa base, o Hamiltoniano do modelo ANNNI admite uma representação em bloco diagonal. A escolha dos operadores de simetria considerados determina o número de setores em que o espaço de Hilbert é particionado.

A Figura 6 apresenta, como exemplo ilustrativo, a representação bloco-diagonal do Hamiltoniano do modelo ANNNI, construída a partir das bases definidas nas Eqs. (94) e (95), para uma cadeia de 4 sítios. A base utilizada esta explicitada a seguir:

$$
\begin{aligned}
|0,0,0,0\rangle &= \frac{1}{\sqrt{2}}(|\uparrow\uparrow\uparrow\uparrow\rangle + |\downarrow\downarrow\downarrow\downarrow\rangle) \\
|1,0,0,0\rangle &= \frac{1}{\sqrt{8}}(|\downarrow\downarrow\uparrow\downarrow\rangle + |\uparrow\uparrow\downarrow\uparrow\rangle + |\uparrow\downarrow\downarrow\downarrow\rangle + |\downarrow\uparrow\uparrow\uparrow\rangle + |\downarrow\uparrow\downarrow\downarrow\rangle + |\uparrow\downarrow\uparrow\uparrow\rangle + |\downarrow\downarrow\downarrow\uparrow\rangle + |\uparrow\uparrow\uparrow\downarrow\rangle) \\
|2,0,0,0\rangle &= \frac{1}{2}(|\uparrow\uparrow\downarrow\downarrow\rangle + |\downarrow\downarrow\uparrow\uparrow\rangle + |\uparrow\downarrow\downarrow\uparrow\rangle + |\downarrow\uparrow\uparrow\downarrow\rangle) \\
|3,0,0,0\rangle &= \frac{1}{\sqrt{2}}(|\uparrow\downarrow\uparrow\downarrow\rangle + |\downarrow\uparrow\downarrow\uparrow\rangle) \\
|0,1,0,0\rangle &= \frac{1}{\sqrt{2}}(|\uparrow\uparrow\uparrow\uparrow\rangle - |\downarrow\downarrow\downarrow\downarrow\rangle) \\
|1,1,0,0\rangle &= \frac{1}{\sqrt{8}}(|\uparrow\uparrow\uparrow\downarrow\rangle + |\uparrow\uparrow\downarrow\uparrow\rangle + |\uparrow\downarrow\uparrow\uparrow\rangle - |\uparrow\downarrow\downarrow\downarrow\rangle + |\downarrow\uparrow\uparrow\uparrow\rangle - |\downarrow\uparrow\downarrow\downarrow\rangle - |\downarrow\downarrow\uparrow\downarrow\rangle - |\downarrow\downarrow\downarrow\uparrow\rangle) \\
|0,0,1,-\rangle &= \frac{1}{\sqrt{8}}(|\uparrow\uparrow\uparrow\downarrow\rangle - i\,|\uparrow\uparrow\downarrow\uparrow\rangle - |\uparrow\downarrow\uparrow\uparrow\rangle + i\,|\uparrow\downarrow\downarrow\downarrow\rangle + i\,|\downarrow\uparrow\uparrow\uparrow\rangle - |\downarrow\uparrow\downarrow\downarrow\rangle - i\,|\downarrow\downarrow\uparrow\downarrow\rangle + |\downarrow\downarrow\downarrow\uparrow\rangle) \\
|0,1,1,-\rangle &= \frac{1}{\sqrt{8}}(|\uparrow\uparrow\uparrow\downarrow\rangle - i\,|\uparrow\uparrow\downarrow\uparrow\rangle - |\uparrow\downarrow\uparrow\uparrow\rangle - i\,|\uparrow\downarrow\downarrow\downarrow\rangle + i\,|\downarrow\uparrow\uparrow\uparrow\rangle + |\downarrow\uparrow\downarrow\downarrow\rangle + i\,|\downarrow\downarrow\uparrow\downarrow\rangle - |\downarrow\downarrow\downarrow\uparrow\rangle) \\
|1,1,1,-\rangle &= \frac{1}{2}(|\uparrow\uparrow\downarrow\downarrow\rangle - i\,|\uparrow\downarrow\downarrow\uparrow\rangle + i\,|\downarrow\uparrow\uparrow\downarrow\rangle - |\downarrow\downarrow\uparrow\uparrow\rangle) \\
|0,0,2,0\rangle &= \frac{1}{2}(|\uparrow\uparrow\downarrow\downarrow\rangle - |\uparrow\downarrow\downarrow\uparrow\rangle - |\downarrow\uparrow\uparrow\downarrow\rangle + |\downarrow\downarrow\uparrow\uparrow\rangle) \\
|0,0,2,1\rangle &= \frac{1}{\sqrt{8}}(|\uparrow\uparrow\uparrow\downarrow\rangle - |\uparrow\uparrow\downarrow\uparrow\rangle + |\uparrow\downarrow\uparrow\uparrow\rangle - |\uparrow\downarrow\downarrow\downarrow\rangle + |\downarrow\uparrow\uparrow\uparrow\rangle + |\downarrow\uparrow\downarrow\downarrow\rangle - |\downarrow\downarrow\uparrow\downarrow\rangle - |\downarrow\downarrow\downarrow\uparrow\rangle) \\
|0,1,2,1\rangle &= \frac{1}{\sqrt{8}}(|\uparrow\uparrow\uparrow\downarrow\rangle - |\uparrow\uparrow\downarrow\uparrow\rangle + |\uparrow\downarrow\uparrow\uparrow\rangle + |\uparrow\downarrow\downarrow\downarrow\rangle - |\downarrow\uparrow\uparrow\uparrow\rangle - |\downarrow\uparrow\downarrow\downarrow\rangle + |\downarrow\downarrow\uparrow\downarrow\rangle - |\downarrow\downarrow\downarrow\uparrow\rangle) \\
|1,1,2,1\rangle &= \frac{1}{\sqrt{2}}(|\uparrow\downarrow\uparrow\downarrow\rangle - |\downarrow\uparrow\downarrow\uparrow\rangle) \\
|0,0,3,-\rangle &= \frac{1}{\sqrt{8}}(|\uparrow\uparrow\uparrow\downarrow\rangle + i\,|\uparrow\uparrow\downarrow\uparrow\rangle - |\uparrow\downarrow\uparrow\uparrow\rangle - i\,|\uparrow\downarrow\downarrow\downarrow\rangle - i\,|\downarrow\uparrow\uparrow\uparrow\rangle - |\downarrow\uparrow\downarrow\downarrow\rangle + i\,|\downarrow\downarrow\uparrow\downarrow\rangle + |\downarrow\downarrow\downarrow\uparrow\rangle) \\
|0,1,3,-\rangle &= \frac{1}{\sqrt{8}}(|\uparrow\uparrow\uparrow\downarrow\rangle + i\,|\uparrow\uparrow\downarrow\uparrow\rangle - |\uparrow\downarrow\uparrow\uparrow\rangle + i\,|\uparrow\downarrow\downarrow\downarrow\rangle - i\,|\downarrow\uparrow\uparrow\uparrow\rangle - |\downarrow\uparrow\downarrow\downarrow\rangle - i\,|\downarrow\downarrow\uparrow\downarrow\rangle - |\downarrow\downarrow\downarrow\uparrow\rangle) \\
|1,1,3,-\rangle &= \frac{1}{2}(|\uparrow\uparrow\downarrow\downarrow\rangle + i\,|\uparrow\downarrow\downarrow\uparrow\rangle - i\,|\downarrow\uparrow\uparrow\downarrow\rangle - |\downarrow\downarrow\uparrow\uparrow\rangle)
\end{aligned}
\tag{96}
$$

Cada bloco é destacado individualmente e identificado pelos respectivos números quânticos que o caracterizam.

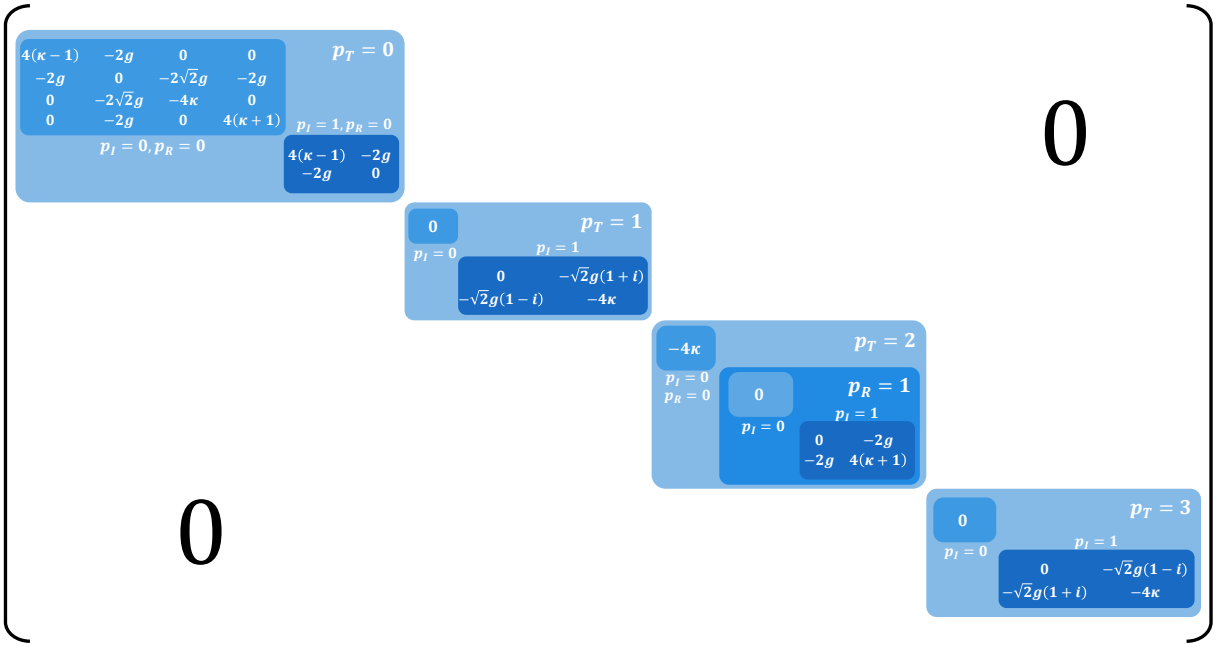

Figura 6 – Representação bloco-diagonal do Hamiltoniano do modelo ANNNI para uma cadeia de 4 sítios, construída a partir dos autovetores comuns à translação e à inversão. Em cada bloco, são indicados os números quânticos p_T e p_I que o caracterizam. Nos setores com número quântico de translação 0 e 2, o Hamiltoniano foi ainda decomposto em blocos menores, utilizando os autoestados comuns à reflexão, sendo os blocos identificados também pelos números quânticos p_R.

7 Agrupamentos de Camadas em FQAs

Neste capítulo, é apresentada uma técnica original desenvolvida no âmbito desta dissertação, sem base em estratégias pré-existentes, com o objetivo de reduzir a profundidade dos circuitos gerados pelo FALQON. A técnica consiste em agrupar camadas consecutivas da evolução temporal em uma única operação efetiva, utilizando uma aproximação baseada na identidade de Baker-Campbell-Hausdorff (BCH) (para mais detalhes ver Apêndice B). Com isso, reescrevemos o produto de exponenciais como uma única exponencial com operadores combinados. Consideramos duas versões da técnica: a LGA-FQA-1, que utiliza uma aproximação de primeira ordem, e a LGA-FQA-2, que incorpora o termo de segunda ordem associado ao comutador entre os Hamiltonianos.

7.1 Agrupamento de Camadas com Aproximação de Primeira Ordem

Nesta seção, consideramos o agrupamento de camadas do FQA utilizando uma aproximação de primeira ordem em relação a Δt. A motivação é simplificar o circuito sem introduzir operadores adicionais, como comutadores, facilitando sua implementação.

Partimos da evolução não trotterizada descrita por:

$$|\psi_\ell\rangle = e^{-i\Delta t(H_p+\beta_\ell H_d)} \cdots e^{-i\Delta t(H_p+\beta_2 H_d)} e^{-i\Delta t(H_p+\beta_1 H_d)} |\psi_0\rangle \,. \tag{97}$$

Considerando que Δt é pequeno, utilizamos a seguinte aproximação:

$$e^{-i\Delta t(H_p+\beta_2 H_d)} e^{-i\Delta t(H_p+\beta_1 H_d)} \approx e^{-i\Delta t[2H_p+(\beta_1+\beta_2)H_d]}. \tag{98}$$

Reescrevemos o termo à direita como uma nova camada com Hamiltoniano efetivo:

$$H_{\text{eff}}^{(2)} = \delta_2 H_p + \alpha_2 H_d, \tag{99}$$

com os parâmetros definidos por:

$$\delta_2 = 2, \quad \alpha_2 = \beta_1 + \beta_2. \tag{100}$$

Esse procedimento pode ser repetido iterativamente. Ao incluir a terceira camada, temos:

$$e^{-i\Delta t(H_p+\beta_3 H_d)}e^{-i\Delta t H_{\text{eff}}^{(2)}} \approx e^{-i\Delta t[(\delta_2+1)H_p+(\alpha_2+\beta_3)H_d]}. \tag{101}$$

Identificamos os parâmetros da terceira camada agrupada como:

$$\delta_3 = \delta_2 + 1, \quad \alpha_3 = \alpha_2 + \beta_3. \tag{102}$$

De forma geral, após o agrupamento de n camadas, obtemos a evolução aproximada:

$$|\psi_\ell\rangle \approx e^{-i\Delta t(H_p+\beta_\ell H_d)} \cdots e^{-i\Delta t(\delta_n H_p+\alpha_n H_d)} |\psi_0\rangle , \tag{103}$$

onde os parâmetros são atualizados segundo as regras:

$$\delta_n = \delta_{n-1} + 1, \quad \alpha_n = \alpha_{n-1} + \beta_n. \tag{104}$$

7.2 Agrupamento de Camadas com Aproximação de Segunda Ordem

Nesta seção, introduzimos uma versão aprimorada da técnica de agrupamento de camadas, incorporando uma aproximação de segunda ordem em Δt. O objetivo principal é capturar correções adicionais causadas pela não comutatividade entre os termos H_p e H_d, o que resulta em uma evolução agrupada mais fiel ao circuito original, especialmente em cenários de alta profundidade.

Começamos com a composição de duas camadas consecutivas da forma:

$$e^{-i\Delta t(H_p+\beta_2 H_d)}e^{-i\Delta t(H_p+\beta_1 H_d)} \approx e^{-i\Delta t[2H_p+(\beta_2+\beta_1)H_d]-\frac{\Delta t^2}{2}(\beta_2-\beta_1)[H_d,H_p]}. \tag{105}$$

Reescrevendo com as variáveis $\alpha_2 = \beta_2 + \beta_1$, $\bar{\gamma}_2 = \beta_2 - \beta_1$ e $\delta_2 = 2$, a expressão torna-se:

$$e^{-i\Delta t[\delta_2 H_p+\alpha_2 H_d]-\frac{\Delta t^2}{2}\bar{\gamma}_2[H_d,H_p]}. \tag{106}$$

Assim, a equação para a evolução no FQA pode ser reescrita como:

$$|\psi_\ell\rangle \approx e^{-i\Delta t(H_p+\beta_\ell H_d)} \cdots e^{-i\Delta t[\delta_2 H_p+\alpha_2 H_d]-\frac{\Delta t^2}{2}\bar{\gamma}_2[H_d,H_p]} |\psi_0\rangle . \tag{107}$$

onde $|\psi_0\rangle$ é o estado inicial.

Agora, repetindo o processo para a terceira camada, obtemos:

$$
\begin{aligned}
|\psi_\ell\rangle &\approx e^{-i\Delta t(H_p+\beta_\ell H_d)} \cdots e^{-i\Delta t[(\delta_2+1)H_p+(\beta_3+\alpha_2)H_d]-\frac{\Delta t^2}{2}(\delta_2\beta_3-\alpha_2+\bar{\gamma}_2)[H_d,H_p]+\mathcal{O}(\Delta t^3)} |\psi_0\rangle \\
&\approx e^{-i\Delta t(H_p+\beta_\ell H_d)} \cdots e^{-i\Delta t[\delta_3 H_p+\alpha_3 H_d]-\frac{\Delta t^2}{2}\bar{\gamma}_3[H_d,H_p]} |\psi_0\rangle ,
\end{aligned}
\tag{108}
$$

onde os novos coeficientes são definidos como:

$$
\alpha_3 = \beta_3 + \alpha_2, \quad \bar{\gamma}_3 = \delta_2\beta_3 - \alpha_2 + \bar{\gamma}_2, \quad \delta_3 = \delta_2 + 1. \tag{109}
$$

Iterando esse processo para n camadas, obtemos a fórmula geral para o agrupamento de n camadas consecutivas. A evolução aproximada do estado é dada por:

$$
\begin{aligned}
|\psi_\ell\rangle &\approx e^{-i\Delta t(H_p+\beta_\ell H_d)} \cdots e^{-i\Delta t[\delta_n H_p+(\beta_n+\alpha_{n-1})H_d]-\frac{\Delta t^2}{2}[\delta_{n-1}\beta_n-\alpha_{n-1}+\bar{\gamma}_{n-1}][H_d,H_p]} |\psi_0\rangle \\
&\approx e^{-i\Delta t(H_p+\beta_\ell H_d)} \cdots e^{-i\Delta t[\delta_n H_p+\alpha_n H_d]-\frac{\Delta t^2}{2}\bar{\gamma}_n[H_d,H_p]} |\psi_0\rangle .
\end{aligned}
\tag{110}
$$

Nesse caso, as variáveis α_n, $\bar{\gamma}_n$ e δ_n podem ser calculadas iterativamente a partir das relações de agrupamento:

$$
\alpha_n = \beta_n + \alpha_{n-1}, \quad \bar{\gamma}_n = \delta_{n-1}\beta_n - \alpha_{n-1} + \bar{\gamma}_{n-1}, \quad \delta_n = \delta_{n-1} + 1. \tag{111}
$$

Para facilitar a implementação do agrupamento de camadas em computadores quânticos, é necessário decompor a exponencial que representa as camadas agrupadas em um produto de três exponenciais. A forma desejada é:

$$
e^{atH_d} e^{btH_p} e^{ct^2[H_d,H_p]}. \tag{112}
$$

Para obter essa decomposição, utilizamos uma aproximação de segunda ordem da identidade de Baker-Campbell-Hausdorff (BCH) e multiplicamos ambos os lados por $e^{\frac{c}{2}\Delta t^2[A,B]}$:

$$
\begin{aligned}
e^{a\Delta tA} e^{b\Delta tB} e^{\frac{c}{2}\Delta t^2[A,B]} &\approx e^{a\Delta tA+b\Delta tB+\frac{ab\Delta t^2}{2}[A,B]} e^{\frac{c}{2}\Delta t^2[A,B]} \\
&\approx e^{a\Delta tA+b\Delta tB+\frac{c+ab}{2}\Delta t^2[A,B]}.
\end{aligned}
\tag{113}
$$

No nosso caso, temos $a = -i\alpha_n$, $b = -i\delta_n$, e o termo c é dado por:

$$
-\bar{\gamma}_n = c + ab = c - \alpha_n\delta_n \quad \Rightarrow \quad c = \alpha_n\delta_n - \bar{\gamma}_n = \gamma_n. \tag{114}
$$

Portanto, a expressão para a decomposição das exponenciais se torna:

$$
\begin{aligned}
e^{-i\Delta t(\delta_n H_p+\alpha_n H_d)-\frac{\Delta t^2}{2}\bar{\gamma}_n[H_d,H_p]} &\approx e^{-i\alpha_n H_d\Delta t} e^{-i\delta_n H_p\Delta t} e^{\frac{\alpha_n\delta_n-\bar{\gamma}_n}{2}[H_d,H_p]\Delta t^2} \\
&\approx U_d(\alpha_n) U_p(\delta_n) U_{com}(\gamma_n),
\end{aligned}
\tag{115}
$$

onde $U_{com}(\gamma_n) = e^{\frac{\gamma_n}{2}[H_d,H_p]\Delta t^2} = e^{-i\frac{\gamma_n}{2}(i[H_d,H_p])\Delta t^2}$ representa o operador de evolução temporal associado ao comutador $i[H_d, H_p]$ em um intervalo de tempo Δt^2, e a regra de atualização dos parâmetros α_n, δ_n, e γ_n é dada por:

$$\begin{aligned} \alpha_n &= \beta_n + \alpha_{n-1} \\ &= -A_{n-1} + \alpha_{n-1}, \end{aligned} \tag{116}$$

$$\delta_n = \delta_{n-1} + 1, \tag{117}$$

$$\gamma_n = \alpha_n \delta_n - \bar{\gamma}_n, \tag{118}$$

onde $\bar{\gamma}_n$ é dado por:

$$\begin{aligned} \bar{\gamma}_n &= \delta_{n-1}\beta_n - \alpha_{n-1} + \bar{\gamma}_{n-1} \\ &= -\delta_{n-1}A_{n-1} - \alpha_{n-1} + \gamma_{n-1}. \end{aligned} \tag{119}$$

7.3 LGA-FQA

A partir das técnicas de agrupamento de camadas, definimos o algoritmo LGA-FQA (Layer Grouping Approach for FQA): LGA-FQA-1, baseado em uma aproximação de primeira ordem para o agrupamento, e LGA-FQA-2, que incorpora uma aproximação de segunda ordem.

7.3.1 LGA-FQA-1

O algoritmo inicia com o estado $|\psi_0\rangle$ e, na primeira iteração, aplica os operadores $U_d(\alpha_1)U_p(\delta_1)$, com os parâmetros inicializados como $\alpha_1 = \beta_1$ e $\delta_1 = 1$.

Durante as $n-1$ iterações seguintes, nenhuma nova camada é adicionada. Em cada iteração k, os parâmetros α_{k-1} e δ_{k-1} são atualizados para novos valores α_k e δ_k, com base no valor de A_{k-1} calculado anteriormente. Assim, uma única camada é mantida e refinada ao longo dessas iterações, simulando a evolução contínua do sistema.

Na iteração $n+1$, uma nova camada agrupada é inserida, composta pelos operadores $U_d(\alpha_{n+1})U_p(\delta_{n+1})$, com os parâmetros reinicializados como $\alpha_{n+1} = \beta_{n+1}$ e $\delta_{n+1} = 1$. Esse padrão de execução se repete: a cada bloco de n iterações, apenas os parâmetros são ajustados, e uma nova camada é adicionada ao final do bloco.

Ao fim de ℓ iterações, o circuito terá aplicado $\lceil \ell/n \rceil$ camadas agrupadas. O estado final do sistema é dado por:

$$|\psi_\ell\rangle = \underbrace{U_d(\alpha_\ell)U_p(\delta_\ell)\ \ldots\ U_d(\alpha_{2n})U_p(\delta_{2n})\ U_d(\alpha_n)U_p(\delta_n)}_{\lceil \ell/n \rceil \text{ camadas}} |\psi_0\rangle \tag{120}$$

A Figura 7 apresenta um diagrama esquemático da estrutura do LGA-FQA-1.

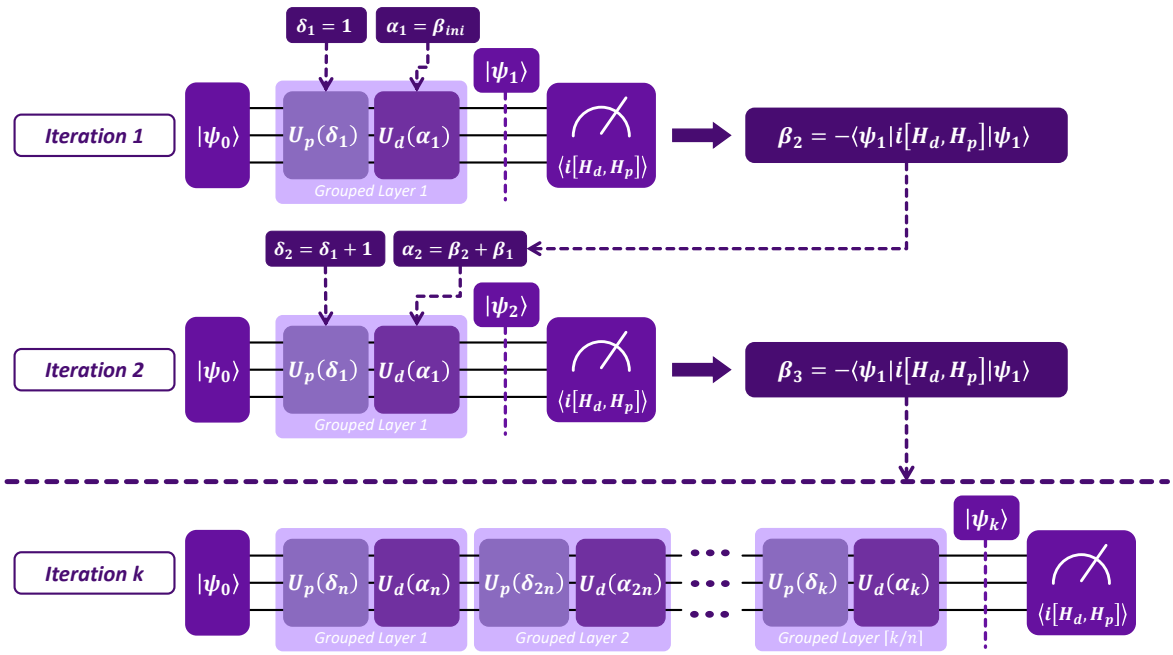

Figura 7 – Diagrama ilustrativo do LGA-FQA-1. O algoritmo inicia com o estado $|\psi_0\rangle$ e aplica, em cada iteração, a mesma camada unitária composta por $U_d(\alpha_k)$ e $U_p(\delta_k)$, com parâmetros α_k e δ_k atualizados adaptativamente. A cada bloco de n iterações, uma nova camada é inserida no circuito, enquanto as camadas anteriores permanecem fixas.

7.3.2 LGA-FQA-2

O LGA-FQA-2 segue a mesma lógica do LGA-FQA-1, com a diferença de que a camada agrupada inclui também o operador U_{com}, o que permite incorporar termos adicionais associados à aproximação de segunda ordem.

O algoritmo parte do estado inicial $|\psi_0\rangle$. Na primeira iteração, aplica-se a sequência de operadores $U_d(\alpha_1)U_p(\delta_1)U_{com}(\gamma_1)$, com os parâmetros inicializados como $\alpha_1 = \beta_1$, $\gamma_1 = \beta_1$ e $\delta_1 = 1$.

Nas $n - 1$ iterações seguintes, a estrutura do circuito permanece inalterada, com uma única camada ativa. Apenas os parâmetros α_k, γ_k e δ_k são atualizados a cada iteração, refinando a evolução implementada por essa camada.

Na iteração $n + 1$, uma nova camada é inserida, composta pelos operadores $U_d(\alpha_{n+1})$ $U_p(\delta_{n+1})$ $U_{com}(\gamma_{n+1})$, com parâmetros reinicializados como $\alpha_{n+1} = \beta_{n+1}$, $\gamma_{n+1} = \beta_{n+1}$ e $\delta_{n+1} = 1$. Esse padrão se repete: a cada bloco de n iterações, os parâmetros das camadas existentes são atualizados e, ao final do bloco, uma nova camada é adicionada.

Após ℓ iterações, o circuito conterá $\lceil \ell/n \rceil$ camadas agrupadas, e o estado final será dado por:

$$|\psi_\ell\rangle = \underbrace{U_d(\alpha_\ell)U_p(\delta_\ell)U_{com}(\gamma_\ell)\ \ldots\ U_d(\alpha_{2n})U_p(\delta_{2n})U_{com}(\gamma_{2n})\ U_d(\alpha_n)U_p(\delta_n)U_{com}(\gamma_n)}_{\lceil \ell/n \rceil \text{ camadas}} |\psi_0\rangle \tag{121}$$

A Figura 8 ilustra a estrutura do LGA-FQA-2.

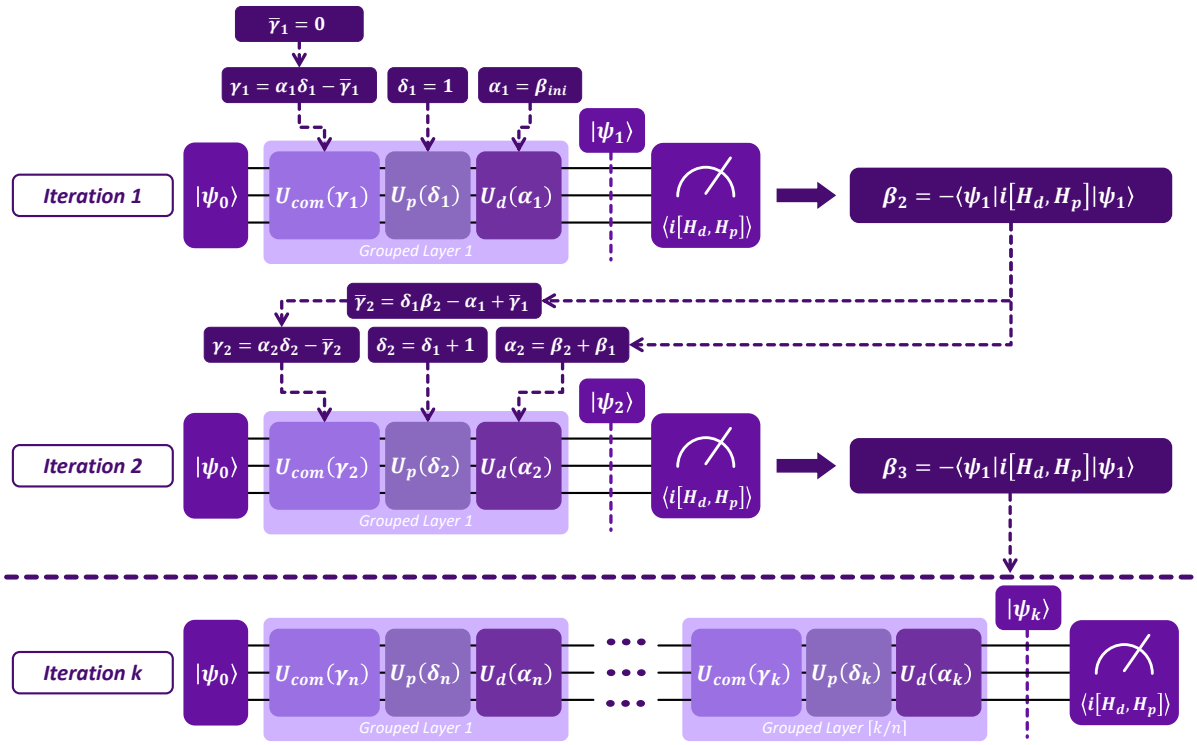

Figura 8 – Diagrama ilustrativo do LGA-FQA-2. O algoritmo começa com o estado inicial $|\psi_0\rangle$ e, em cada iteração, aplica uma camada composta pelos operadores $U_d(\alpha_k)$, $U_p(\delta_k)$ e $U_{com}(\gamma_k)$, com os parâmetros α_k, δ_k e γ_k atualizados de forma adaptativa. A cada bloco de n iterações, uma nova camada agrupada é inserida, enquanto as camadas anteriores permanecem fixas com os últimos parâmetros que possuíram.

Vale ressaltar que a precisão da aproximação depende diretamente do valor de Δt: quanto menor esse intervalo, mais fiel será a representação da evolução contínua desejada. Entretanto, intervalos menores demandam mais iterações para simular um tempo fixo, o que pode aumentar o custo computacional. É necessário, portanto, ajustar adequadamente o número de camadas agrupadas em relação ao intervalo de tempo adotado.

7.4 Comparação entre Profundidades

Um ponto relevante a ser analisado é a diferença de profundidade entre uma camada padrão do FQA e uma camada agrupada. No caso do LGA-FQA-1, a camada agrupada apresenta a mesma profundidade de uma camada convencional do FQA, já que não há operadores adicionais. Em contraste, no LGA-FQA-2, a presença do operador U_{com}, associado ao comutador entre os Hamiltonianos H_d e H_p, aumenta a profundidade da camada agrupada.

Para ilustrar esse aumento de profundidade, consideremos problemas de otimização combinatória, geralmente codificados em Hamiltonianos do tipo Ising que descrevem uma cadeia de L sítios. Nesse contexto, o Hamiltoniano do problema é dado por:

$$H_p = \sum_{i=1}^{L} \mu_i \sigma_i^z + \sum_{i=1}^{L} \sum_{j>i}^{L} \omega_{i,j} \sigma_i^z \sigma_j^z, \tag{122}$$

cujo operador de evolução temporal pode ser escrito como

$$U_p(\Delta t) = \prod_{i=1}^{L} e^{-i\Delta t \mu_i \sigma_i^z} \prod_{i=1}^{L} \prod_{j>i}^{L} e^{-i\Delta t \omega_{i,j} \sigma_i^z \sigma_j^z}. \tag{123}$$

Considerando que a implementação de $e^{-i\Delta t \sigma_i^z}$ requer um conjunto de portas de profundidade d_1, e que $e^{-i\Delta t \sigma_i^z \sigma_j^z}$ demanda uma profundidade d_2, podemos analisar o impacto desses termos na profundidade total do circuito. O primeiro produto, envolvendo apenas termos locais, pode ser implementado com profundidade d_1, já que os operadores σ_i^z comutam entre si, permitindo que suas evoluções sejam aplicadas em paralelo. Em contraste, o segundo produto, envolvendo interações de dois qubits, é mais restritivo. No pior caso, considerando que os termos não possam ser paralelizados, a profundidade necessária será $d_2 \frac{L(L-1)}{2}$, correspondente ao número de interações distintas entre pares de qubits.

O Hamiltoniano driver, por sua vez, é tipicamente escolhido como

$$H_d = \sum_{i=1}^{L} \sigma_i^x, \tag{124}$$

com o operador de evolução temporal associado dado por

$$U_d(\Delta t) = \prod_{i=1}^{L} e^{-i\Delta t \sigma_i^x}. \tag{125}$$

Como os termos σ_i^x também comutam entre si, todos os operadores podem ser aplicados em paralelo, resultando em uma profundidade d_3, correspondente à profundidade necessária para implementar uma única rotação $e^{-i\Delta t \sigma_i^x}$. Assim, a profundidade total de uma camada padrão do FQA será $d_p = d_1 + d_3 + d_2 \frac{L(L-1)}{2}$.

Ao considerar a camada agrupada, analisamos o comutador entre H_d e H_p, dado por

$$[H_d, H_p] = \sum_{i=1}^{L} -2i\mu_i \sigma_i^y + \sum_{i=1}^{L} \sum_{\substack{j=1 \\ j \neq i}}^{L} -2i\omega_{i,j} \sigma_i^y \sigma_j^z, \tag{126}$$

cujo operador de evolução temporal assume a forma

$$U_{com}(\Delta t) = \prod_{i=1}^{L} e^{-2i\Delta t \mu_i \sigma_i^y} \prod_{i=1}^{L} \prod_{\substack{j=1 \\ j \neq i}}^{L} e^{-2i\Delta t \omega_{i,j} \sigma_i^y \sigma_j^z}. \tag{127}$$

Definindo d_4 e d_5 como as profundidades necessárias para implementar $e^{-2i\Delta t \sigma_i^y}$ e $e^{-2\Delta t \sigma_i^y \sigma_j^z}$, respectivamente, a profundidade total para implementar U_{com} será $d_4 + d_5 L(L-1)$. Esse resultado decorre do mesmo raciocínio usado na análise de U_p, considerando, no pior caso, a impossibilidade de paralelização das interações de dois qubits.

Assim, a profundidade de uma camada agrupada é dada por:

$$d_a = d_1 + d_3 + d_4 + \left(\frac{d_2}{2} + d_5\right) L(L-1). \tag{128}$$

Para valores grandes de L, a razão entre a profundidade da camada agrupada e a profundidade da camada padrão converge para:

$$1 + \frac{2d_5}{d_2}. \tag{129}$$

Assumindo que $d_5 = d_2$, o que é razoável considerando que as portas de dois qubits associadas a $\sigma_i^y \sigma_j^z$ possuem complexidade comparável às associadas a $\sigma_i^z \sigma_j^z$, concluímos que a profundidade de uma camada agrupada no LGA-FQA-2 é aproximadamente três vezes a profundidade de uma camada convencional.

8 Algoritmo Quântico Baseado em Feedback com Reescalonamento Temporal

Neste capítulo, apresentamos uma adaptação do algoritmo FQA, baseada no reescalonamento do tempo, denominada "*Time-Rescaled Feedback-Based Quantum Algorithm*" (TR-FQA). Essa técnica foi desenvolvida no âmbito desta dissertação em colaboração com Guilherme Pexe, e deu origem a um artigo submetido para avaliação na *Physical Review A*, com um preprint disponibilizado no arXiv (55). Discutimos como incorporar essa abordagem no processo de realimentação do FQA, por meio da redefinição da função de controle e do ajuste da evolução temporal. O objetivo é investigar como o reescalonamento pode influenciar a convergência do algoritmo e propor uma formulação que mantenha a simplicidade de implementação em circuitos quânticos.

8.1 Método de Reescalonamento Temporal

O Método de Reescalonamento Temporal foi introduzido por Bernado em (56) e posteriormente desenvolvido por Ferreira et al. em (57). Esse método modifica a dependência temporal do Hamiltoniano de um sistema quântico, alterando sua dinâmica de evolução e permitindo que o estado final desejado seja alcançado de forma mais eficiente, seja acelerando ou desacelerando o processo conforme necessário.

Para formalizar essa ideia, considera-se um sistema quântico governado por um Hamiltoniano dependente do tempo $H(t)$, evoluindo no intervalo $t \in [t_i = 0, t_f]$ e descrito pela equação de Schrödinger. A evolução unitária correspondente é dada por:

$$U\left(t_f\right) = \mathcal{T} \exp\left(-i \int_0^{t_f} H\left(t'\right) dt'\right). \tag{130}$$

A dinâmica do sistema pode ser modificada através de uma função de reescalonamento $t = f(\tau)$, que redefine a evolução temporal. Aplicando essa mudança de variável, a evolução unitária é reescrita como:

$$\mathcal{U}\left(\tau_f\right) = \mathcal{T} \exp\left(-i \int_{f^{-1}(0)}^{f^{-1}\left(t_f\right)} H(f(\tau))\dot{f}(\tau) d\tau\right). \tag{131}$$

Essa expressão descreve a evolução de um sistema governado pelo Hamiltoniano reescalonado:

$$\mathcal{H}(\tau) = H(f(\tau))\dot{f}(\tau), \tag{132}$$

que depende da função de reescalonamento $f(\tau)$ e de sua derivada em relação a τ, $\dot{f}(\tau)$. Dessa forma, se a evolução do sistema reescalonado ocorrer no intervalo $\tau \in [\tau_i = f^{-1}(0), \tau_f =$

$f^{-1}(t_f)]$, ela produzirá o mesmo estado final $|\psi(t_f)\rangle$ que a evolução original de $H(t)$, desde que ambas partam do mesmo estado inicial $|\psi(0)\rangle$.

Isso implica que, escolhendo adequadamente a função $f(\tau)$, o processo governado por $\mathcal{H}(\tau)$ pode ser acelerado ($\tau_f - \tau_i < t_f - t_i$) ou desacelerado ($\tau_f - \tau_i > t_f - t_i$) em relação à evolução original. Caso a evolução original corresponda a uma transformação adiabática, e a função $f(\tau)$ seja escolhida de modo que: (i) os tempos iniciais coincidam, ou seja, $f^{-1}(0) = 0$; (ii) o tempo final seja reduzido, $f^{-1}(t_f) < t_f$; e (iii) os Hamiltonianos inicial e final permaneçam inalterados, $\mathcal{H}(0) = H(0)$ e $\mathcal{H}(\tau_f) = H(t_f)$, o método de reescalonamento temporal pode ser classificado como um atalho para adiabaticidade.

Uma escolha comum para a função de reescalonamento é:

$$f_1(\tau) = a\tau - \frac{t_f}{2\pi a}(a-1)\sin\left(\frac{2\pi a}{t_f}\tau\right), \tag{133}$$

onde a é um parâmetro que controla a contração temporal. Outra possibilidade é uma função polinomial:

$$f_2(\tau) = \frac{2(a^2 - a^3)}{t_f^2}\tau^3 + \frac{3(a^2 - a)}{t_f}\tau^2 + \tau. \tag{134}$$

Essas funções permitem modificar a evolução temporal do sistema sem alterar seu estado final, com o tempo total de evolução dado por $\Delta\tau = \Delta t/a$.

Tendo descrito a ideia central do método de reescalonamento temporal, a próxima etapa é adaptá-lo ao contexto do FQA. Essa adaptação modifica a dinâmica de evolução do algoritmo, aumentando sua eficiência e permitindo que a solução desejada seja alcançada em menor tempo computacional.

8.2 TR-FQA

Baseando-nos no método de reescalonamento temporal, podemos adaptá-lo para uso com o FQA. Para isso, consideramos a equação de Schrödinger aplicada a um Hamiltoniano dependente do tempo da forma $H(t) = H_p + \beta(t)H_d$. Ao reescalonarmos o tempo por meio da transformação $t = f(\tau)$, a Eq. (1) torna-se:

$$\begin{aligned} \frac{d}{d\tau}|\psi(\tau)\rangle &= -i\mathcal{H}(\tau)|\psi(\tau)\rangle, \\ &= -iH(f(\tau))\dot{f}(\tau)|\psi(\tau)\rangle. \end{aligned} \tag{135}$$

Para obtermos a função de controle nesse caso, seguimos os mesmos passos utilizados no FQA. Primeiro, calculamos a derivada temporal do valor esperado de H_p ao longo da

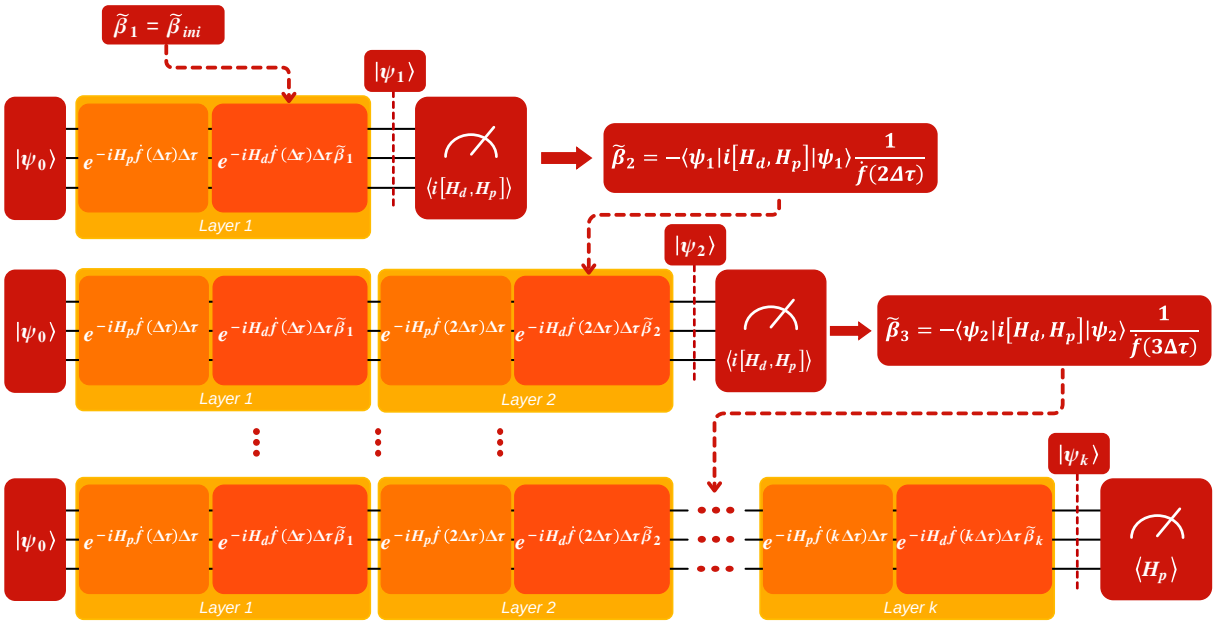

Figura 9 – Diagrama ilustrativo do TR-FQA. O processo começa com o estado inicial $|\psi_0\rangle$. Em cada camada k, os operadores unitários $e^{-iH_p\dot{f}(k\Delta\tau)\Delta\tau}$ e $e^{-iH_d\dot{f}(k\Delta\tau)\Delta\tau\tilde{\beta}_k}$ são aplicados sequencialmente, ajustando-se de forma adaptativa o parâmetro $\tilde{\beta}_k$.

evolução reescalonada:

$$\begin{aligned}\frac{d}{d\tau}\langle\psi(\tau)|\,H_p\,|\psi(\tau)\rangle &= i\,\langle\psi(\tau)|\,[\mathcal{H}(\tau), H_p]\,|\psi(\tau)\rangle\,,\\ &= i\dot{f}(\tau)\,\langle\psi(\tau)|\,[H(f(\tau)), H_p]\,|\psi(\tau)\rangle\,,\\ &= \beta(f(\tau))\dot{f}(\tau)\,\langle\psi(\tau)|\,i[H_d, H_p]\,|\psi(\tau)\rangle\,,\\ &= \tilde{\beta}(\tau)A(\tau)\dot{f}(\tau),\end{aligned} \tag{136}$$

onde definimos $A(\tau) = \langle\psi(\tau)|i[H_d, H_p]|\psi(\tau)\rangle$ e $\tilde{\beta}(\tau) = \beta(f(\tau))$, que representa a função de controle avaliada no tempo reescalonado.

Para garantirmos que a função custo, definida como o valor esperado de H_p, diminua ao longo da evolução, determinamos uma função de controle $\tilde{\beta}(\tau)$ que satisfaça:

$$\frac{d}{d\tau}\langle\psi(\tau)|\,H_p\,|\psi(\tau)\rangle \leq 0, \quad \forall\tau. \tag{137}$$

Uma forma geral de satisfazer essa desigualdade é adotar:

$$\tilde{\beta}(\tau) = -\omega\,F(\tau, A(\tau))\,G(\dot{f}(\tau)), \tag{138}$$

onde $\omega > 0$, $F(\tau, A(\tau))$ é uma função contínua com $F(\tau, 0) = 0$ e $A(\tau)F(\tau, A(\tau)) \geq 0$ para todo $A(\tau) \neq 0$, e $G(\dot{f}(\tau))$ é uma função contínua com $G(\dot{f}(\tau))\dot{f}(\tau) \geq 0$.

Neste trabalho, adotamos $\omega = 1$, $F(\tau, A(\tau)) = A(\tau)$ e $G(\dot{f}(\tau)) = 1/\dot{f}(\tau)$, de modo que $\tilde{\beta}(\tau)$ torna-se:

$$\tilde{\beta}(\tau) = -A(\tau)\frac{1}{\dot{f}(\tau)}. \tag{139}$$

Para que essa expressão seja bem definida, exige-se que $\dot{f}(\tau) \neq 0$ em todo o domínio considerado.

Agora, podemos adaptar o circuito do FQA para incorporar o tempo reescalonado. Para isso, consideramos a evolução temporal reescalonada no intervalo de $\tau_0 = 0$ até $\tau_f = f^{-1}(t_f)$, discretizado em k passos com intervalo de tempo $\Delta\tau$. Essa evolução pode ser descrita por uma sequência de aplicações do operador unitário:

$$\mathcal{U}(\tau) = \exp\Big(-i(H_p + \tilde{\beta}(\tau)H_d)\dot{f}(\tau)\Delta\tau\Big). \tag{140}$$

A aplicação de $\mathcal{U}(\tau)$ sobre o estado $|\psi(\tau)\rangle$ produz o estado atualizado $|\psi(\tau + \Delta\tau)\rangle$. Para implementarmos $\mathcal{U}(\tau)$ em um circuito quântico, utilizamos uma aproximação trotterizada, o que nos permite expressar $\mathcal{U}(\tau)$ como:

$$\mathcal{U}(\tau) = \mathcal{U}_d(\tilde{\beta}(\tau))\mathcal{U}_p, \tag{141}$$

onde $\mathcal{U}_d(\tilde{\beta}(\tau)) = \exp\Big(-iH_d\tilde{\beta}(\tau)\dot{f}(\tau)\Delta\tau\Big)$ e $\mathcal{U}_p = \exp\Big(-iH_p\dot{f}(\tau)\Delta\tau\Big)$. A partir dessa decomposição, descrevemos o estado resultante do circuito FQA após uma sequência de k aplicações desses operadores trotterizados:

$$|\psi_k\rangle = \mathcal{U}_d(\tilde{\beta}_k)\mathcal{U}_p \;\; \ldots \;\; \mathcal{U}_d(\tilde{\beta}_2)\mathcal{U}_p\, \mathcal{U}_d(\tilde{\beta}_1)\mathcal{U}_p\, |\psi_0\rangle\,, \tag{142}$$

onde $|\psi_k\rangle$ é o estado do circuito após a aplicação da k-ésima camada ($|\psi_k\rangle \approx |\psi(k\Delta\tau)\rangle$) e $\tilde{\beta}_k$ é o parâmetro de controle da k-ésima camada ($\tilde{\beta}_k \approx \tilde{\beta}(k\Delta\tau)$).

Além disso, de forma análoga ao FQA padrão, como $\tilde{\beta}_k$ é necessário previamente para construirmos o estado $|\psi_k\rangle$, utilizamos o estado da iteração anterior. Para intervalos de tempo suficientemente pequenos, esse estado aproxima bem $|\psi_k\rangle$, permitindo que definamos a lei de realimentação como:

$$\tilde{\beta}_k = -A_{k-1}\,\frac{1}{\dot{f}(k\Delta\tau)} = -\,\langle\psi_{k-1}|\,H_p\,|\psi_{k-1}\rangle\,\frac{1}{\dot{f}(k\Delta\tau)}. \tag{143}$$

Dessa forma, conseguimos adaptar o FQA para incorporar o reescalonamento do tempo. A Figura 9 ilustra o funcionamento do TR-FQA. A estrutura dos operadores permanece idêntica à do FQA, sem exigir modificações adicionais na implementação. Além disso, a nova lei de feedback difere da original apenas por um fator $1/\dot{f}(\tau)$, o que preserva sua forma funcional e dispensa medições extras. Como a função de reescalonamento depende dos parâmetros a e t_f, o desempenho do TR-FQA também é sensível a esses valores, que devem ser escolhidos com atenção para garantir bons resultados.

9 Metodologia

Este trabalho foi conduzido em três etapas principais: (i) estudo da influência das simetrias na convergência do algoritmo FQA; (ii) desenvolvimento de estratégias para redução da profundidade dos circuitos, com a implementação do LGA-FQA e LGA-FALQON; e (iii) modificação da dinâmica do algoritmo por meio de reescalonamento temporal, resultando no TR-FQA e TR-FALQON.

Na etapa (i), a análise concentrou-se exclusivamente na preparação de estados quânticos de sistemas físicos, utilizando o modelo ANNNI como caso de teste, com o Hamiltoniano problema definido pela Eq. (84). Os autoestados de spin de σ_z, $|\uparrow\rangle$ e $|\downarrow\rangle$ foram mapeados para os estados $|0\rangle$ e $|1\rangle$, respectivamente. O objetivo foi investigar como as simetrias do sistema influenciam a convergência do FQA. Para isso, foram considerados diferentes estados iniciais, a fim de explorar a influência dessa escolha sobre a convergência do algoritmo.

Nas etapas (ii) e (iii), os algoritmos foram avaliados tanto na preparação de estados quanto na resolução de problemas de otimização. Para a preparação de estados físicos, foi novamente utilizado o modelo ANNNI, com o algoritmo FQA em suas versões modificadas (LGA-FQA e TR-FQA). Para os problemas de otimização, foi adotado o problema MaxCut, utilizando o algoritmo FALQON em suas versões correspondentes (LGA-FALQON e TR-FALQON). O Hamiltoniano problema para o MaxCut é dado pela Eq. (60).

Em todas as etapas, o Hamiltoniano de condução adotado foi:

$$H_d = \sum_{j=1}^{L} \sigma_j^x, \tag{144}$$

e, nas etapas (ii) e (iii), o estado inicial foi mantido fixo como:

$$|\psi_0\rangle = \prod_{i=1}^{L} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \tag{145}$$

onde L representa o número de qubits necessários para codificar o problema: o número de vértices no grafo (para o MaxCut) ou o número de sítios na cadeia (para o modelo ANNNI).

Nos casos em que o modelo ANNNI foi utilizado como caso teste, explorou-se o fato de que seu Hamiltoniano pode ser reescrito como $H_p = H_{zz} - gH_d$, onde

$$H_{zz} = -\sum_{j=0}^{L-1} \left(\sigma_j^z \sigma_{j+1}^z - \kappa\, \sigma_j^z \sigma_{j+2}^z\right) \tag{146}$$

representa os termos de interação de primeiro e segundo vizinhos. Com essa decomposição, a camada de evolução padrão do algoritmo, originalmente dada por $U_d(\beta_k)U_p$, foi implementada

como:

$$U_d(\beta_k - g)\, U_{zz} = e^{-i\Delta t(\beta_k - g)H_d}\, e^{-i\Delta t H_{zz}}. \tag{147}$$

Essa reformulação permitiu reutilizar diretamente a rotina de aplicação de U_d, bastando ajustar o parâmetro de rotação β_k para absorver o termo proporcional a g.

Todas as análises foram realizadas por meio de simulações numéricas, com algoritmos implementados em Python a partir de código próprio. As simulações foram executadas no Google Colab, ambiente de computação em nuvem com acesso gratuito a recursos de hardware. O sistema utilizado foi o Ubuntu 22.04.4 LTS (Jammy Jellyfish), com Python 3.11.11 e o framework PyTorch 2.5.1+cu124 e NumPy 2.0.2. O ambiente contava com um processador Intel(R) Xeon(R) 2.20GHz, 12 GB de RAM, 108 GB de armazenamento e uma GPU Tesla T4 compatível com CUDA 12.4, empregada para acelerar as operações matriciais necessárias nas simulações de sistemas de qubits grandes.

9.1 Influência das Simetrias no Desempenho do FQA

A influência das simetrias na convergência do FQA foi investigada com base no fato de que o algoritmo depende da aplicação de operadores de evolução temporal construídos a partir de Hamiltonianos não comutantes. Quando o Hamiltoniano dependente do tempo, $H(t) = H_p + \beta(t)H_d$, preserva certas simetrias, ou seja, quando H_p e H_d compartilham as mesmas simetrias, o estado do sistema permanece no setor correspondente ao estado inicial ao longo de toda a execução do algoritmo.

Consequentemente, se o FQA for iniciado em um setor diferente daquele que contém o estado fundamental de H_p, o algoritmo não será capaz de alcançá-lo, pois as simetrias preservadas restringem a evolução do sistema a transições dentro do mesmo setor. Esse comportamento limita a capacidade do FQA de explorar completamente o espaço de Hilbert, mas também permite direcionar o algoritmo para a preparação de estados excitados ao se iniciar a evolução em setores distintos do setor fundamental.

Para investigar esses efeitos, utilizamos o modelo ANNNI como Hamiltoniano problema, $H_p = H_A(\kappa, g)$, como já mencionado anteriormente, com simulações realizadas em cadeias de $L = 4$ sítios. Foram considerados diferentes estados iniciais com simetrias específicas, caracterizadas pelos números quânticos de inversão p_I, translação p_T e reflexão p_R, conforme descrito no Capítulo 6.

Essa análise foi realizada em colaboração com Guilherme Pexe e resultou na publicação do artigo (58) na *Physical Review B*, no qual utilizamos as simetrias do sistema para determinar o diagrama de fases do modelo ANNNI.

Inicialmente, analisamos a convergência do FQA ao se partir de vetores de base de setores definidos, conforme:

$$|j, p_I, p_T, p_R\rangle$$

onde j é um índice que percorre os vetores da base ortonormal do setor correspondente. Os estados utilizados estão listados em (96).

Também foram utilizados como estados iniciais os autoestados de $-H_d$ com números quânticos bem definidos, representados por:

$$|\chi_i, p_I, p_T, p_R\rangle$$

em que χ_i denota o i-ésimo autoestado de menor energia de $-H_d$.

Em uma etapa posterior, avaliamos a dinâmica do FQA quando iniciado em uma superposição de estados pertencentes a setores distintos:

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}\left(|j, p_I, p_T, p_R\rangle + |j', p'_I, p'_T, p'_R\rangle\right)$$

permitindo explorar o impacto da quebra intencional de simetrias na convergência.

Por fim, examinamos a aplicação do WFQAE para a extração do espectro de um setor específico. Para isso, $p+1$ circuitos foram inicializados com uma base ortonormal do setor desejado:

$$\begin{aligned}
\left|\phi_0^{(0)}\right\rangle &= |0, p_I, p_T, p_R\rangle \\
\left|\phi_0^{(1)}\right\rangle &= |1, p_I, p_T, p_R\rangle \\
&\vdots \\
\left|\phi_0^{(p)}\right\rangle &= |p, p_I, p_T, p_R\rangle
\end{aligned}$$

Essa abordagem busca investigar o comportamento do WFQAE em contextos nos quais o espectro apresenta degenerescência, mas os estados degenerados estão distribuídos em diferentes setores de simetria. Ao restringir a análise a um único setor, espera-se que o algoritmo possa isolar parte do espectro e recuperar autovalores distintos, contornando parcialmente sua limitação frente a espectros degenerados.

9.2 Impacto do Agrupamento de Camadas

Nesta etapa, propôs-se uma técnica para agrupar camadas do algoritmo FQA, visando reduzir a profundidade do circuito quântico sem comprometer significativamente a qualidade da

solução. A estratégia consiste em reescrever a sequência de operadores de evolução usando uma aproximação baseada na identidade de Baker-Campbell-Hausdorff, e foi denominada LGA-FQA (*Layer Grouping Approach for FQA*), conforme descrito no Capítulo 7.

Foram consideradas duas variantes: LGA-FALQON-1, que utiliza apenas o termo linear da expansão, e LGA-FALQON-2, que inclui o primeiro comutador entre os operadores.

As abordagens foram comparadas ao FALQON convencional para avaliar a capacidade de preparar estados próximos ao ótimo em problemas de otimização. A análise inicial focou no problema MaxCut, utilizando um grafo ponderado 3-regular com 16 vértices gerado aleatoriamente.

Os seguintes aspectos foram analisados:

- Fidelidade entre os estados finais obtidos pelas três abordagens;
- Robustez do agrupamento em relação à variação de Δt;
- Impacto do número de camadas agrupadas (n) sobre o desempenho;
- Profundidade total dos circuitos produzidos por cada método.

Em uma segunda fase, avaliou-se a capacidade do LGA-FQA-1 e LGA-FQA-2 em preparar o estado fundamental do modelo ANNNI em uma cadeia de 8 sítios, comparando os resultados com os obtidos pelo FQA convencional. O foco esteve na fidelidade do estado final e na redução da profundidade do circuito.

9.3 Efeito do Reescalonamento Temporal na Convergência do FQA

Nesta etapa, propomos a aplicação do método TR-FQA, que adapta o algoritmo FQA para incorporar um reescalonamento temporal na evolução do sistema. O objetivo do TR-FQA é melhorar a convergência e a eficiência do FQA, ajustando dinamicamente a função de controle ao longo do processo de evolução, como detalhado no Capítulo 8.

Na primeira etapa, comparou-se o TR-FALQON com o FALQON convencional, considerando os seguintes aspectos:

- A fidelidade dos estados finais obtidos pelo TR-FALQON em comparação com o FALQON convencional;
- A robustez do TR-FALQON frente às variações nos parâmetros a e t_f, que controlam o comportamento do reescalonamento temporal;
- O impacto do reescalonamento temporal na profundidade do circuito e no tempo total de evolução do sistema;

- A capacidade do TR-FALQON de gerar estados cuja solução do problema seja claramente evidenciada.

Para essa análise, utilizamos o problema MaxCut, em um grafo ponderado 3-regular com 16 vértices.

Em seguida, avaliou-se a capacidade do TR-FQA para preparar o estado fundamental do modelo ANNNI, utilizando uma cadeia de 12 sítios. Os resultados obtidos pelo TR-FQA foram comparados com os do FQA convencional, com ênfase na fidelidade dos estados finais e na eficiência do algoritmo.

10 Resultados

Este capítulo apresenta os resultados obtidos a partir da aplicação das metodologias descritas no Capítulo 9. As análises foram organizadas em três frentes principais: a influência das simetrias do Hamiltoniano na convergência do FQA, a avaliação do impacto do agrupamento de camadas na preparação de estados, e o estudo dos efeitos do reescalonamento temporal no desempenho do algoritmo. Cada uma dessas frentes é discutida em seções dedicadas, com base em simulações numéricas e métricas específicas que permitiram comparar diferentes estratégias e variantes do FQA.

10.1 Efeitos das Simetrias na Convergência do FQA

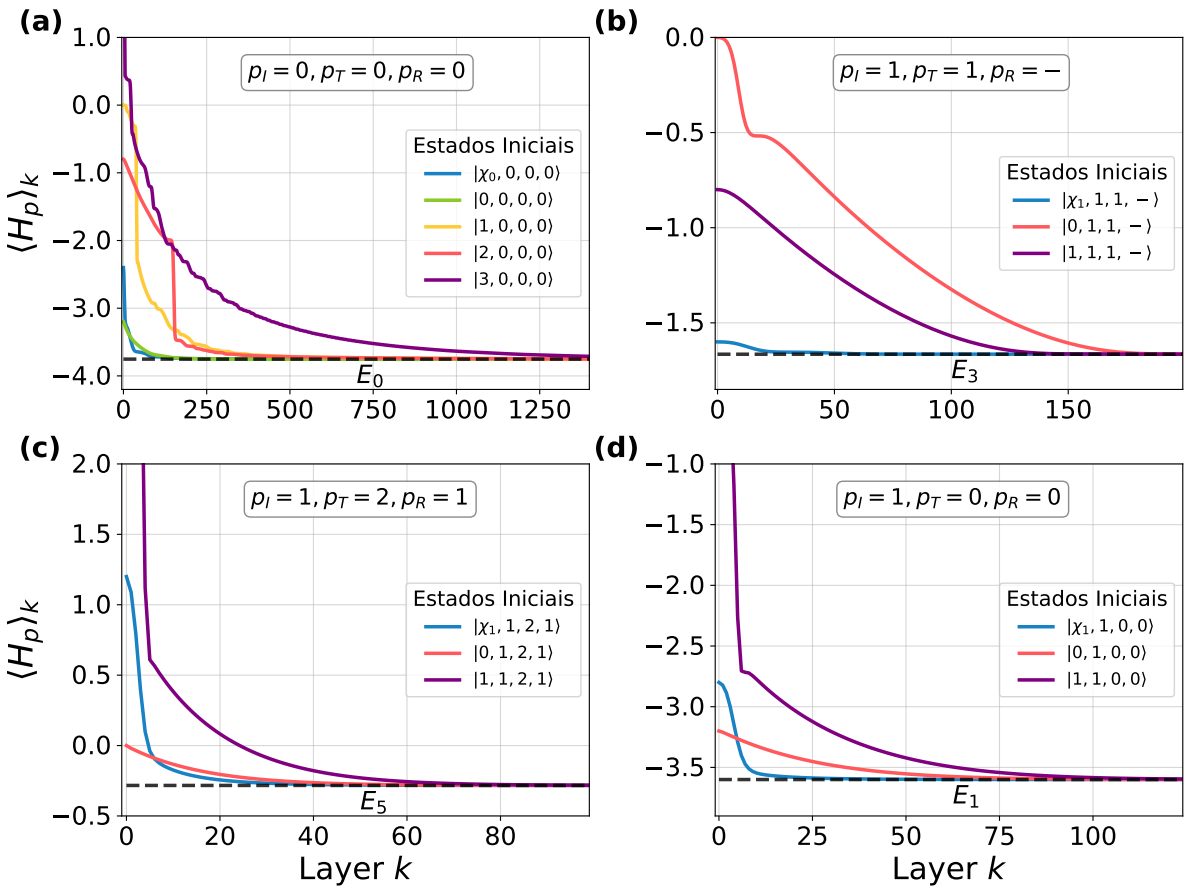

Figura 10 – Convergência do FQA a partir de diferentes estados iniciais pertencentes a setores simétricos distintos no modelo ANNNI com 4 sítios. Cada painel corresponde a um setor de simetria diferente, e a linha tracejada indica a menor energia permitida dentro daquele setor

A Figura 10 apresenta a evolução do valor esperado da energia ao longo das camadas do FQA, considerando diferentes estados iniciais pertencentes ao mesmo setor simétrico. As simulações foram realizadas para o modelo ANNNI com uma cadeia de 4 sítios, utilizando os parâmetros $\kappa = 0,2$, $g = 0,6$ e passo temporal $\Delta t = 0,06$. Cada painel corresponde a um setor de simetria específico, identificado por seus números quânticos p_I, p_T e p_R.

A linha tracejada em cada gráfico representa o menor valor de energia dentro daquele setor, que pode ou não coincidir com o estado fundamental do sistema. Os resultados demonstram que, ao iniciar a evolução com diferentes estados dentro de um mesmo setor, todos convergem para o mesmo nível de energia, indicando que o FQA é capaz de explorar o subespaço simétrico adequadamente. Por exemplo:

1. No painel (a), todos os estados iniciais convergem para o estado fundamental;
2. No painel (b), para o terceiro estado excitado;
3. No painel (c), para o quinto estado excitado;
4. No painel (d), para o primeiro estado excitado.

Apesar da convergência para o mesmo nível de energia dentro de cada setor, observa-se uma diferença no número de camadas necessário para atingir a convergência, a depender da escolha do estado inicial. Por exemplo, no setor $(p_I = 0, p_T = 0, p_R = 0)$, o estado $|\chi_0, 0, 0, 0\rangle$ e o estado $|0, 0, 0, 0\rangle$ alcançam rapidamente a energia mínima do setor, com menos de 200 camadas, enquanto o estado $|3, 0, 0, 0\rangle$ exige mais de 1250 camadas para convergir.

Esse comportamento evidencia dois pontos principais. o FQA preserva as simetrias do estado inicial ao longo da evolução. Isso implica que o algoritmo não é capaz de transitar entre setores distintos. Portanto, iniciar o algoritmo com um estado que não pertença ao setor do estado alvo inviabiliza a preparação desse estado.

Embora todos os estados do mesmo setor eventualmente convirjam para a mesma energia mínima acessível, a taxa de convergência depende da escolha do estado inicial. Assim, mesmo dentro de um setor, a eficiência do algoritmo pode ser otimizada por uma escolha mais adequada do ponto de partida.

Na sequência, investigou-se como a presença de componentes pertencentes a setores de simetria diferentes no estado inicial afeta a convergência do FQA. Para isso, analisou-se o comportamento da energia esperada ao longo da evolução do algoritmo para três tipos de estados iniciais: (i) o estado fundamental de $-H_d$, (ii) um estado excitado pertencente ao mesmo setor que o primeiro estado excitado de H_p, e (iii) a superposição normalizada desses dois estados.

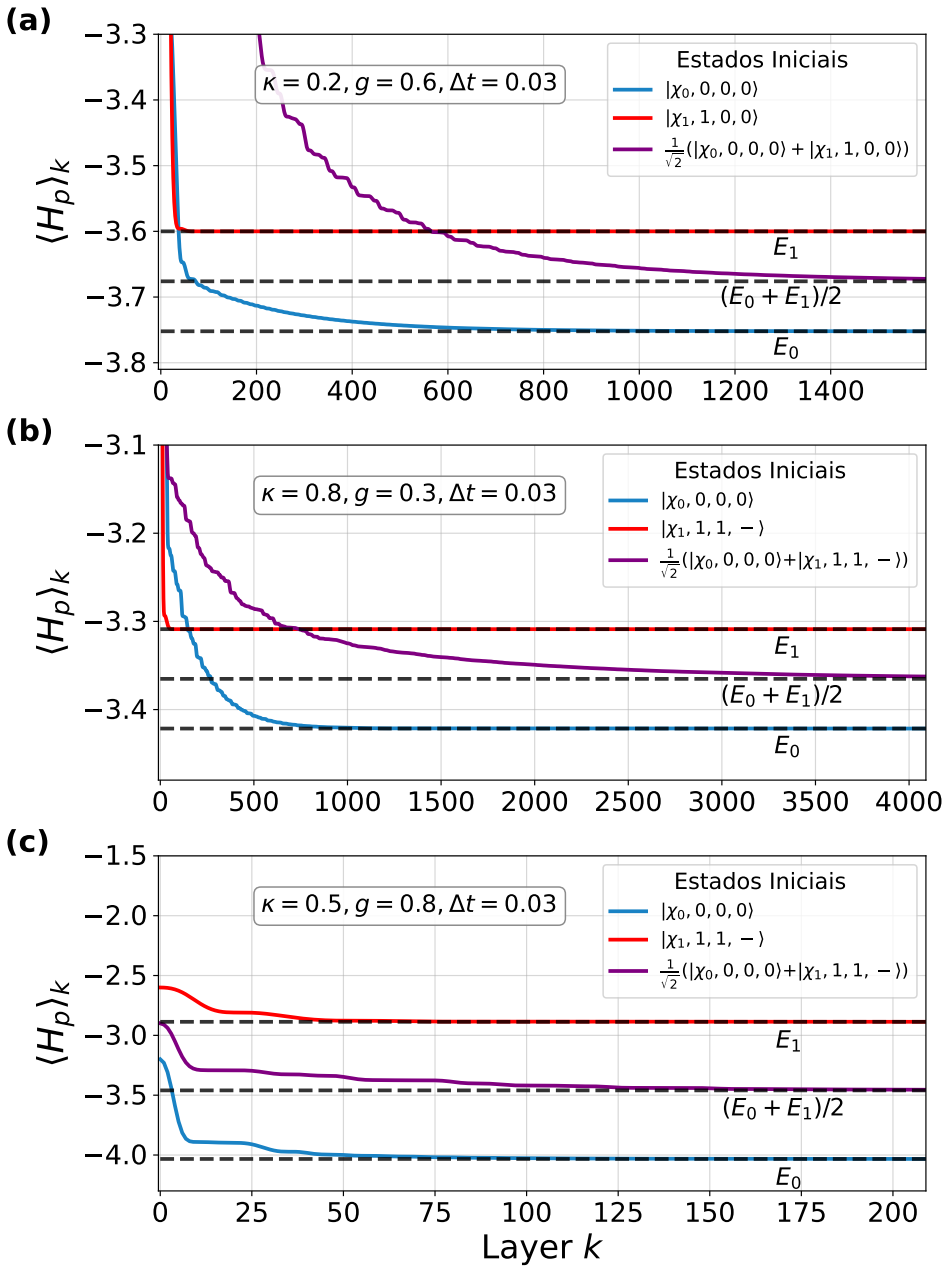

Figura 11 – Convergência do valor esperado da energia $\langle H \rangle_k$ durante a evolução pelo FQA para três estados iniciais distintos: o estado fundamental de $-H_d$, o primeiro estado excitado com simetria compatível com a do primeiro excitado de H_p, e a superposição linear normalizada de ambos. As simulações foram feitas para o modelo ANNNI com diferentes conjuntos de parâmetros κ e g indicados em cada painel, representando três fases distintas do sistema.

A Figura 11 apresenta os resultados dessa análise para o modelo ANNNI com uma cadeia de 4 sítios, utilizando passo de tempo fixo $\Delta t = 0,03$, nas seguintes configurações de parâmetros:

- Painel (a): $\kappa = 0,2$, $g = 0,6$ — fase ferromagnética;
- Painel (b): $\kappa = 0,8$, $g = 0,3$ — antifase;
- Painel (c): $\kappa = 0,5$, $g = 0,8$ — fase paramagnética.

O estado fundamental de $-H_d$ foi mantido fixo em todos os casos como $|\chi_0\rangle = |\chi_0, 0, 0, 0\rangle$, definido na equação (145). Esse estado pertence ao mesmo bloco de simetria do estado fundamental de H_p, independentemente da fase considerada. Para cada configuração, selecionou-se um primeiro estado excitado de $-H_d$ que compartilha as mesmas simetrias do primeiro estado excitado de H_p na respectiva fase. Esses estados são denotados genericamente por $|\chi_1\rangle$, e a superposição utilizada foi dada por:

$$|\psi_{\text{sup}}\rangle = \frac{1}{\sqrt{2}} \left(|\chi_0\rangle + |\chi_1\rangle \right). \tag{148}$$

Os resultados mostram que os estados iniciais $|\chi_0\rangle$ e $|\chi_1\rangle$ convergem, respectivamente, para o estado fundamental e para o primeiro estado excitado de H_p, como esperado, validando a preservação das simetrias na dinâmica gerada pelo FQA. Já a evolução a partir da superposição $|\psi_{\text{sup}}\rangle$ apresenta uma energia final que corresponde, à média das energias dos dois estados individuais. Esse comportamento é consistente com a linearidade do valor esperado da energia, refletindo que, em cada componente, o algoritmo respeita o confinamento setores e conduz cada parte da superposição à energia mínima de seu respectivo setor.

Esses resultados reforçam dois pontos importantes. Primeiro, há preservação de simetrias: a evolução gerada pelo FQA atua separadamente em cada componente da superposição, respeitando as simetrias associadas a cada termo. Não há mistura entre os blocos, e cada parte evolui independentemente dentro de seu setor. Segundo, há uma consequência direta para superposições: a energia final de uma superposição de estados pertencentes a setores diferentes não corresponde, necessariamente, a um estado físico relevante, pois o resultado é apenas a média ponderada das energias finais de cada componente, sem colapso para um estado puramente fundamental ou excitado. Assim, estados iniciais que misturam simetrias podem resultar em preparações não ideais se o objetivo for obter um estado puro com energia bem definida.

Por fim, analisou-se a capacidade do WFQAE em obter autovalores do Hamiltoniano dentro de setores específicos de simetria, para diferentes escolhas de setores iniciais.

A Figura 12 mostra a convergência do valor esperado da energia ao longo da evolução para diferentes estados iniciais do WFQAE. Em cada caso, os estados iniciais foram escolhidos

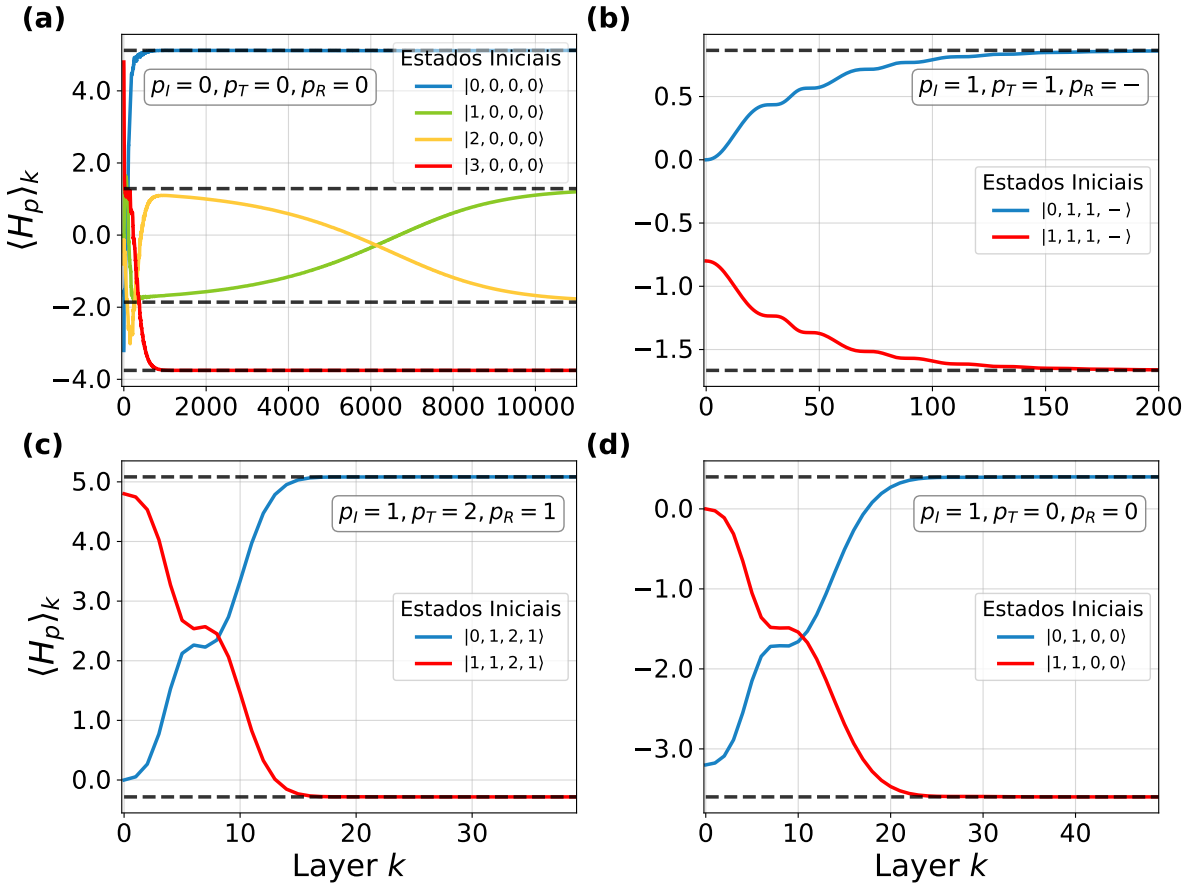

Figura 12 – Convergência do valor esperado da energia ao longo da evolução do WFQAE para diferentes estados iniciais, organizados por setores de simetria. Cada painel refere-se a um setor distinto: (a) $(p_I, p_T, p_R) = (0, 0, 0)$, (b) $(1, 1)$, (c) $(1, 2)$, e (d) $(1, 0, 0)$. Em cada caso, os estados iniciais foram escolhidos da base do respectivo setor, com pesos distintos atribuídos na lei de *feedback*. Observa-se que cada trajetória converge para um autovalor do Hamiltoniano dentro do bloco correspondente ao setor de simetria, evidenciando a capacidade do WFQAE em explorar eficientemente o espectro restrito a cada subespaço.

a partir da base correspondente ao setor de simetria, conforme definido em (96). Os parâmetros utilizados foram $\kappa = 0,2$, $g = 0,6$ e passo de tempo $\Delta t = 0,06$.

No painel (a), os estados iniciais pertencem à base do setor $(p_I, p_T, p_R) = (0, 0, 0)$, o qual possui quatro estados. Os pesos utilizados na lei de *feedback* para esses estados foram $0,1$, $0,4$, $0,7$ e 1. Já nos painéis (b), (c) e (d), os estados iniciais pertencem a setores de dimensão 2, sendo: (b) $(p_I, p_T) = (1, 1)$, (c) $(p_I, p_T) = (1, 2)$ e (d) $(p_I, p_T, p_R) = (1, 0, 0)$. Para esses três setores, foram utilizados dois estados por setor, com pesos $\omega_q = 0,1$ e 1.

Em todos os casos, a lei de *feedback* foi multiplicada por um fator $1/4$ com o objetivo de evitar que o parâmetro β se tornasse excessivamente grande, o que poderia comprometer a estabilidade da convergência.

Os resultados da figura indicam que, em todos os setores analisados, o WFQAE foi capaz de conduzir o valor esperado da energia de cada estado inicial à convergência para um autovalor do Hamiltoniano H_p contido no bloco associado ao respectivo setor de simetria. Nota-se ainda que os estados associados aos menores pesos na função de *feedback* tenderam a convergir para os autovalores mais altos do setor, enquanto os estados com pesos maiores foram direcionados aos autovalores mais baixos, coerentemente com o papel atribuído à função de custo.

Esses resultados demonstram que, ao particionar o espaço de Hilbert em subespaços correspondentes a setores com simetrias bem definidas, e nos quais o espectro de H_p não apresenta degenerescências, é possível aplicar o WFQAE para explorar o espectro completo dentro de cada setor. Dessa forma, contorna-se uma das principais limitações do algoritmo, que é a dificuldade de lidar diretamente com autovalores degenerados, permitindo assim uma extração controlada dos autovalores por meio da evolução independente em cada subespaço.

10.2 LGA-FQA e LGA-FALQON

Nesta seção, são apresentados os estudos realizados com os algoritmos LGA-FQA e LGA-FALQON. As análises foram divididas em duas etapas principais. A primeira investiga o desempenho desses algoritmos na resolução de problemas de otimização. A segunda avalia a eficácia do LGA-FQA na preparação de estados quânticos, utilizando o modelo ANNNI como caso de teste.

A principal métrica de avaliação adotada foi a convergência do valor esperado da energia $\langle H_p \rangle_k$ ao longo das iterações, em função da profundidade acumulada do circuito, denotada por d_k. Para fins de comparação, essa profundidade foi normalizada pela profundidade de uma camada padrão do FQA, representada por d_p. Com isso, analisou-se o desempenho dos algoritmos em termos da profundidade relativa d_k/d_p. É importante destacar que a camada agrupada do LGA-FQA-1 foi considerada equivalente a uma camada padrão (d_p), enquanto a camada do LGA-FQA-2, devido à inclusão de termos de segunda ordem, foi considerada com profundidade três vezes maior ($3d_p$), como explicado na seção 7.4.

Inicialmente, analisou-se o comportamento do LGA-FALQON para diferentes ordens de aproximação (primeira e segunda ordem). As simulações foram realizadas sobre um grafo 3-regular ponderado com 16 vértices. Para fins de comparação, o FALQON foi executado com passo de tempo fixo $\Delta t = 0,04$, valor identificado como próximo do valor crítico (Δt_c) para essa configuração.

A Figura 13 compara o desempenho do FALQON com o LGA-FALQON, considerando as aproximações de primeira e segunda ordem no agrupamento, referidas nos gráficos como 1 (LGA-FQA-1) e 2 (LGA-FQA-2), respectivamente. As simulações do LGA-FALQON foram realizadas com $n = 10$ camadas agrupadas e $\Delta t = 0,02$.

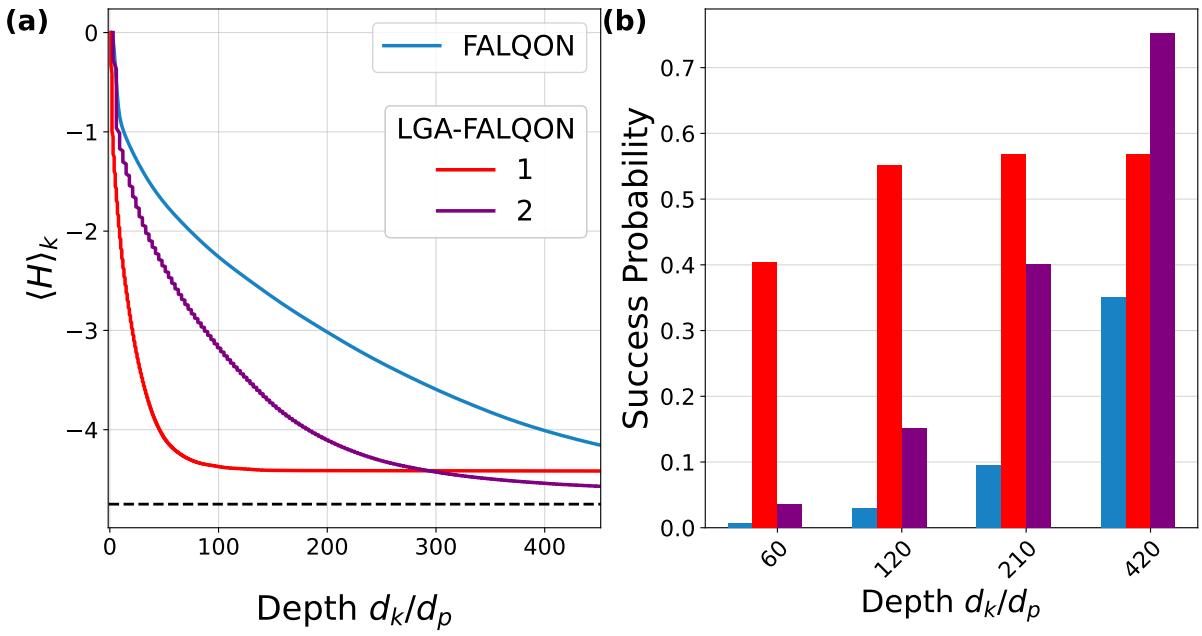

Figura 13 – Simulações numéricas comparando o desempenho dos algoritmos LGA-FQA e FQA. O painel (a) mostra a convergência da função de custo $J = \langle\psi_k|H|\psi_k\rangle$ em função da profundidade relativa do circuito, d_k/d_p, onde d_p corresponde à profundidade de uma camada padrão do FQA. A linha tracejada indica a energia do estado fundamental do Hamiltoniano H_p. O painel (b) apresenta a probabilidade de obtenção de uma solução correta do problema para diferentes profundidades.

Observa-se que o LGA-FALQON, nas duas ordens de aproximação, proporciona uma queda mais rápida no valor esperado da energia $\langle H_p \rangle$ em profundidades menores, quando comparado ao FALQON. Esse efeito é mais evidente para o LGA-FQA-1, que apresenta uma redução inicial mais acentuada devido à menor profundidade de suas camadas. No entanto, essa aproximação tende à estagnação a partir de uma certa profundidade, limitando a melhora na qualidade da solução.

Por outro lado, o LGA-FQA-2, apesar de apresentar uma queda mais lenta com o aumento das camadas, não sofre com estagnação no intervalo observado. Com o aumento da profundidade do circuito, essa versão ultrapassa o desempenho do LGA-FQA-1, superando-o a partir de $d_k/d_p \approx 300$.

O algoritmo FALQON, embora apresente uma convergência mais gradual, também não sofre interrupções no decaimento da energia, mantendo uma redução contínua ao longo das iterações. Ainda assim, o LGA-FALQON foi capaz de atingir bons resultados com profundidades menores.

No painel (b) da Figura 13, observa-se que, com profundidade relativa $d_k/d_p \approx 210$, o LGA-FQA-1 já apresentava uma probabilidade de sucesso superior a 50%, enquanto o LGA-FQA-2 alcançava aproximadamente 40%. Nessa mesma profundidade, o FALQON permaneceu

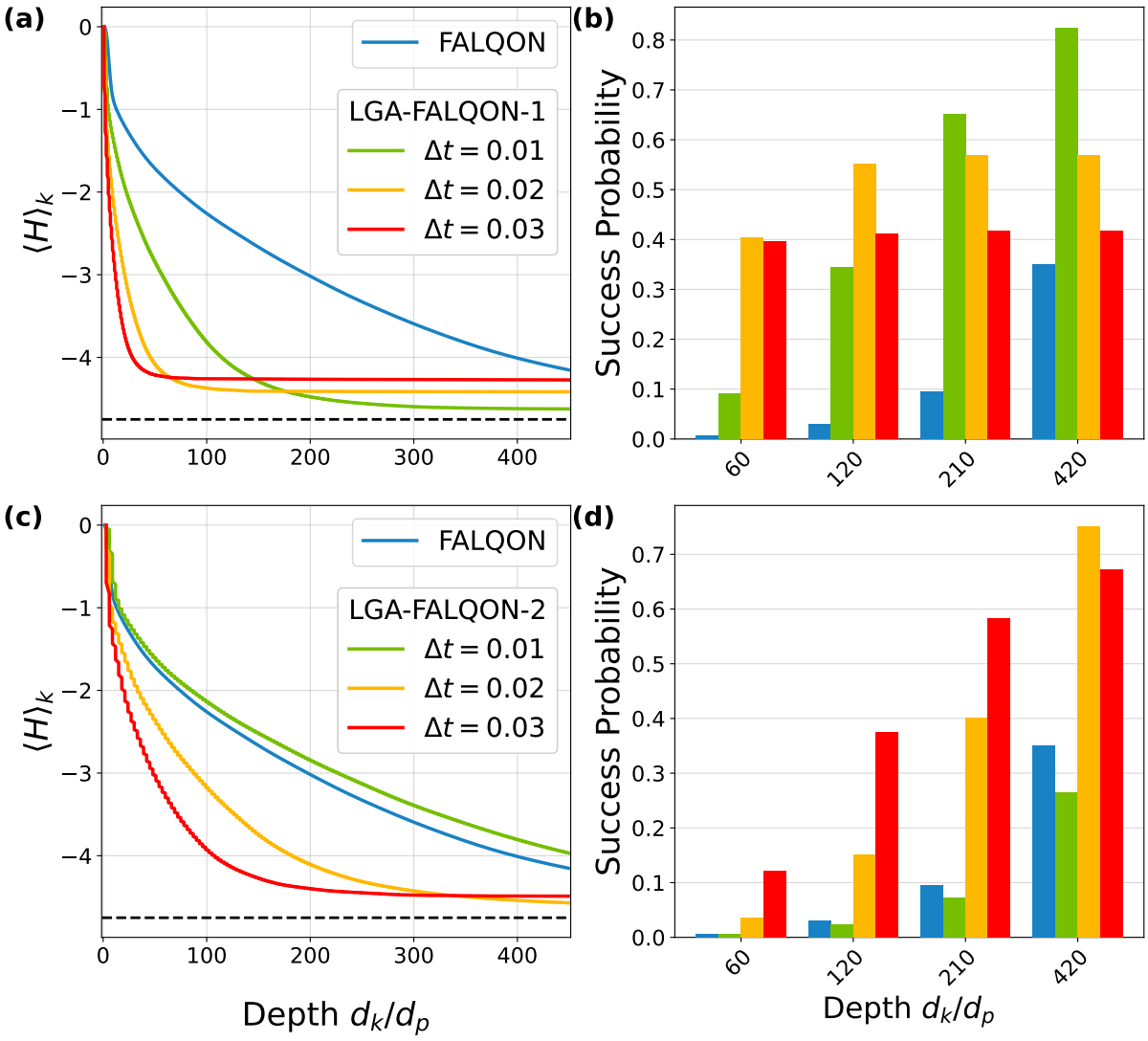

Figura 14 – Comparação entre os algoritmos FQA e LGA-FQA para diferentes valores do passo de tempo Δt, com $n = 10$ camadas agrupadas. Os painéis (a) e (c) apresentam a convergência da função de custo $J = \langle\psi_k|H|\psi_k\rangle$ em função da profundidade relativa do circuito d_k/d_p, enquanto os painéis (b) e (d) mostram a probabilidade de se obter uma solução correta para diferentes profundidades. Os painéis (a) e (b) referem-se ao LGA-FQA-1 (primeira ordem) e os painéis (c) e (d) ao LGA-FQA-2 (segunda ordem). A linha tracejada representa a energia do estado fundamental de H_p.

abaixo de 10%. Já com $d_k/d_p \approx 420$, o LGA-FQA-2 superou o desempenho do LGA-FQA-1, atingindo uma probabilidade de sucesso acima de 70%, enquanto o FALQON atingiu pouco mais de 30%.

Esses resultados indicam que o LGA-FALQON é capaz de evidenciar a solução ótima com profundidade menor que o FALQON convencional, sendo especialmente eficaz em regimes de baixa profundidade de circuito.

Em seguida, analisou-se o comportamento do LGA-FQA variando-se dois de seus principais parâmetros: o número de camadas agrupadas n e o passo de tempo Δt.

A Figura 14 apresenta a comparação entre os algoritmos FQA e LGA-FQA para diferentes valores de Δt, mantendo-se fixo o número de camadas agrupadas $n = 10$. Os painéis (a) e (b) mostram os resultados para a versão de primeira ordem (LGA-FQA-1), enquanto os painéis (c) e (d) referem-se à versão de segunda ordem (LGA-FQA-2).

Em ambos os casos, observa-se que valores maiores de Δt promovem uma convergência mais rápida do valor esperado da energia em relação à profundidade do circuito. No entanto, essa convergência acelerada tende a ser seguida por estagnação precoce. No LGA-FQA-1, por exemplo, com $\Delta t = 0,03$, a estagnação ocorre antes da profundidade 100, com a probabilidade de sucesso estabilizando em torno de 40%. Já para $\Delta t = 0,01$, a queda da energia é mais lenta, mas contínua, superando 80% de probabilidade em profundidade 420. O caso intermediário, $\Delta t = 0,02$, apresenta desempenho intermediário, com estagnação um pouco mais lenta e probabilidade final entre 50% e 60%.

No LGA-FQA-2, a convergência é mais gradual, e a estagnação é menos evidente. Apenas o caso com $\Delta t = 0,03$ apresentou sinais de saturação, por volta da profundidade 400, com probabilidade próxima de 70%. Para $\Delta t = 0,02$, observou-se o melhor desempenho: não houve estagnação no intervalo analisado, e a probabilidade de sucesso em profundidade 420 ultrapassou 70%. Já com $\Delta t = 0,01$, o desempenho foi inferior ao do FQA padrão, indicando que passos de tempo muito pequenos não favorecem a versão de segunda ordem.

A Figura 15 ilustra como o desempenho do LGA-FQA é impactado pelo número de camadas agrupadas n, mantendo-se fixo o passo de tempo em $\Delta t = 0,02$. O parâmetro n controla o número de camadas do FQA agrupadas em uma única camada efetiva.

Nos painéis (a) e (b), referentes ao LGA-FQA-1, observa-se que $n = 15$ proporciona uma convergência rápida nas primeiras profundidades, mas essa vantagem inicial é também limitada por uma estagnação precoce. Em contraste, valores menores de n, como $n = 5$, mantêm uma evolução mais lenta porém sustentada, alcançando desempenhos superiores em profundidades maiores. O caso intermediário, $n = 10$, apresenta um balanço entre esses dois comportamentos.

Para o LGA-FQA-2 (painéis (c) e (d)), o impacto de n segue uma tendência semelhante, embora mais suave. A evolução é mais gradual e a estagnação ocorre de forma menos abrupta.

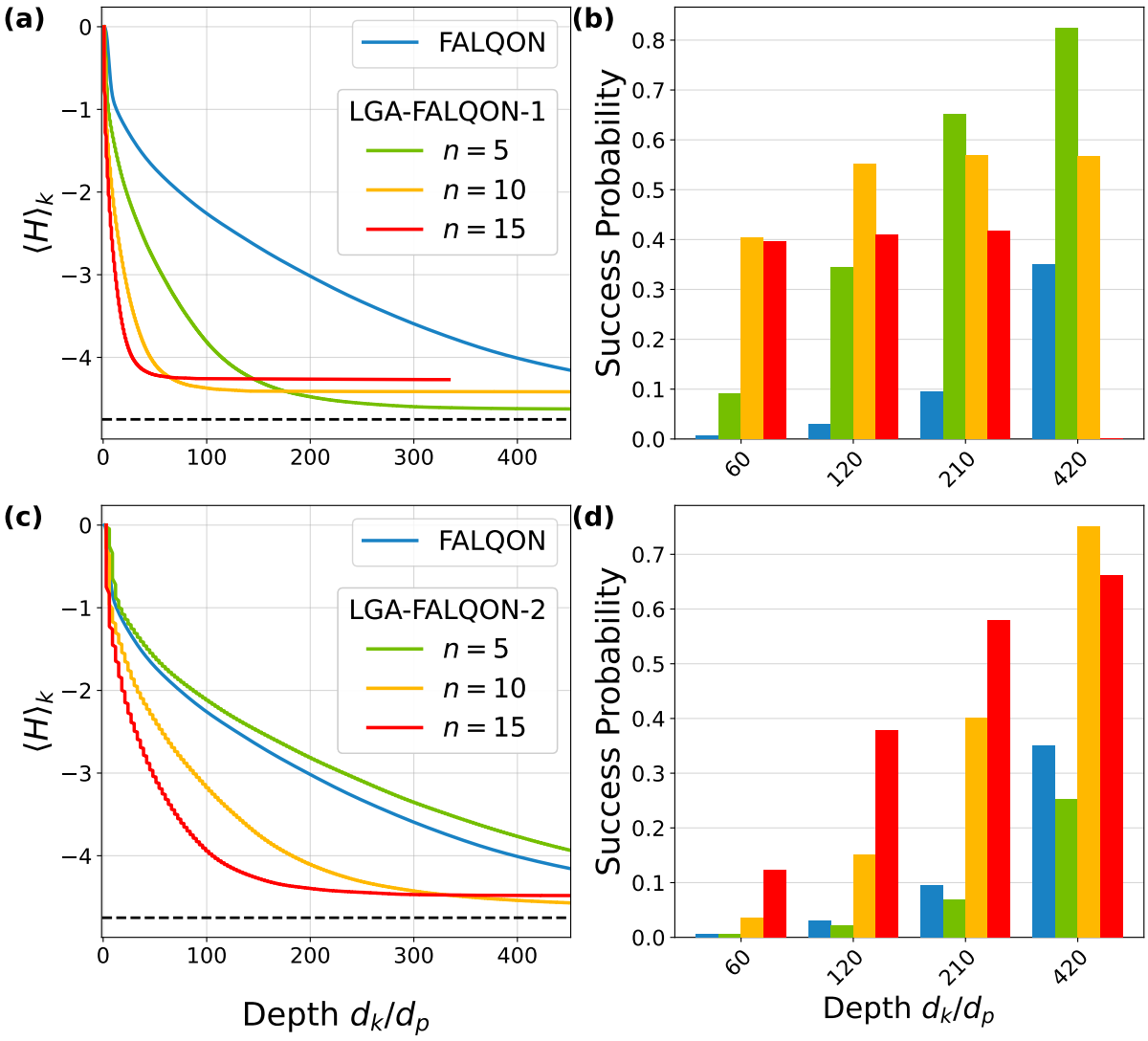

Figura 15 – Comparação entre os algoritmos FQA e LGA-FQA para diferentes números de camadas agrupadas, com passo de tempo fixo $\Delta t = 0{,}02$. Os painéis (a) e (c) apresentam a convergência da função de custo $J = \langle\psi_k|H|\psi_k\rangle$ em função da profundidade relativa do circuito d_k/d_p, enquanto os painéis (b) e (d) mostram a probabilidade de se obter uma solução correta para diferentes profundidades. Os painéis (a) e (b) referem-se ao LGA-FQA-1 (primeira ordem) e os painéis (c) e (d) ao LGA-FQA-2 (segunda ordem). A linha tracejada representa a energia do estado fundamental de H_p.

Destaca-se que, com $n = 10$, essa versão supera o FQA original no intervalo analisado, enquanto $n = 5$ resultou em desempenho inferior ao do FQA original, sugerindo que agrupamentos muito pequenos podem limitar a eficácia da versão de segunda ordem.. Já para $n = 15$, o desempenho é comprometido pela estagnação, embora menos acentuada do que na versão de primeira ordem.

Ao comparar as duas versões propostas, notam-se comportamentos distintos. O LGA-FALQON-1 tende a apresentar quedas mais acentuadas nos valores esperados da energia logo nas primeiras iterações, o que favorece uma identificação mais rápida da solução. Entretanto, a estagnação precoce é mais frequente, especialmente para Δt e n elevados. Já o LGA-FALQON-2 apresenta convergência mais gradual e estável, com desempenho superior em profundidades mais elevadas. Contudo, esse comportamento mais regular é obtido ao custo de uma camada agrupada três vezes mais profunda que a do FALQON padrão, o que pode não ser vantajoso em relação ao FALQON padrão.

Dessa forma, o LGA-FALQON-1 é mais indicado em contextos onde se busca evidenciar rapidamente a solução do problema com circuitos rasos, ainda que com menor probabilidade final de sucesso. Já o LGA-FALQON-2, por apresentar melhor desempenho em profundidades maiores, pode ser mais adequado quando o objetivo é maximizar a qualidade da solução, desde que o orçamento de profundidade permita acomodar suas camadas mais custosas. Em ambas as versões, a necessidade de utilizar valores de Δt menores que os empregados no FQA padrão impõe um aumento no número total de iterações para alcançar convergência, o que deve ser considerado na avaliação do custo computacional total do algoritmo.

Por fim, avaliou-se a capacidade do LGA-FQA em preparar o estado fundamental do modelo ANNNI. Para isso, foram realizadas simulações considerando uma cadeia com 8 sítios e duas configurações distintas dos parâmetros κ e g.

A Figura 16 mostra a convergência dos algoritmos FQA, LGA-FQA-1 e LGA-FQA-2 em função da profundidade relativa do circuito, para os dois conjuntos de parâmetros considerados. Em ambos os casos, as duas versões do LGA-FQA conseguiram aproximar-se da energia do estado fundamental do modelo ANNNI. No entanto, observaram-se diferenças no ritmo de convergência entre as versões.

Apesar de o LGA-FQA-2 poder utilizar um número maior de camadas agrupadas ($n = 20$) e um passo de tempo menor ($\Delta t = 0.0020$), a convergência do LGA-FQA-1 foi mais rápida na prática. Com $n = 15$ e $\Delta t = 0.0015$, o LGA-FQA-1 alcançou valores próximos ao mínimo da função de custo em uma profundidade entorno de 280. Já o LGA-FQA-2, embora também tenha se aproximado da energia do estado fundamental, o fez com uma profundidade significativamente maior, acima de 500.

Esses resultados reforçam a tendência observada anteriormente: o LGA-FQA-1 tende a ser mais eficiente na obtenção de soluções aproximadas com profundidade reduzida, enquanto

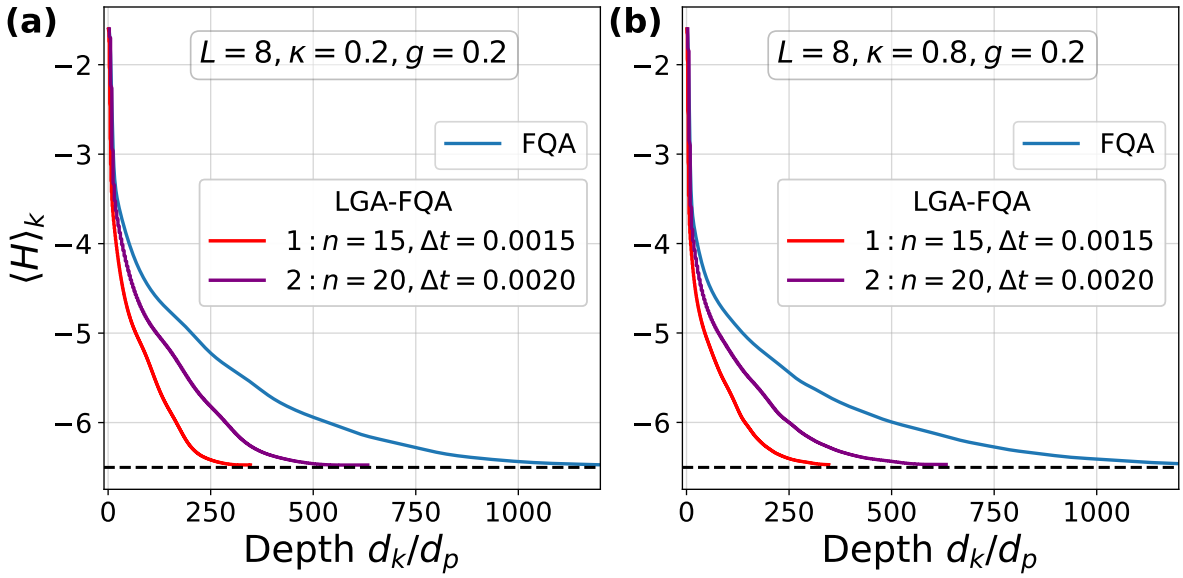

Figura 16 – Simulações numéricas comparando FQA e LGA-FQA aplicados ao modelo ANNNI, com duas configurações de parâmetros para uma cadeia de 8 sítios: (a) $\kappa = 0.2, g = 0.2$ e (b) $\kappa = 0.8, g = 0.2$. As simulações com LGA-FQA utilizaram a abordagem de agrupamento, com n e Δt indicados em cada painel. O FQA foi simulado com $\Delta t = 0.01$. As linhas tracejadas representam a energia do estado fundamental, enquanto as curvas mostram a convergência da função de custo $J = \langle \Psi_k | H | \Psi_k \rangle$ em função da profundidade relativa d_k/d_p.

o LGA-FQA-2 apresenta convergência mais lenta, exigindo circuitos mais profundos para atingir desempenho semelhante.

10.3 TR-FQA e TR-FALQON

Nesta seção, são apresentados os estudos realizados com os algoritmos TR-FQA e TR-FALQON. As análises foram divididas em duas etapas principais. A primeira avalia o desempenho dos algoritmos na resolução de problemas de otimização. A segunda investiga a eficácia do TR-FQA na preparação de estados quânticos, utilizando o modelo ANNNI como caso de teste.

Inicialmente, analisou-se o comportamento do TR-FALQON sob diferentes funções de reescalonamento temporal, bem como a influência dos parâmetros de contração a e tempo final t_f. As simulações foram conduzidas em um grafo 3-regular ponderado com 16 vértices. Tanto o TR-FALQON quanto o FALQON foram executados com passo de tempo fixo $\Delta t = \Delta \tau = 0,04$, valor próximo ao do passo de tempo crítico.

A Figura 17 compara o desempenho do FALQON com o TR-FALQON utilizando duas funções de reescalonamento temporal distintas: f_1, definida na Eq. (133), e f_2, definida na

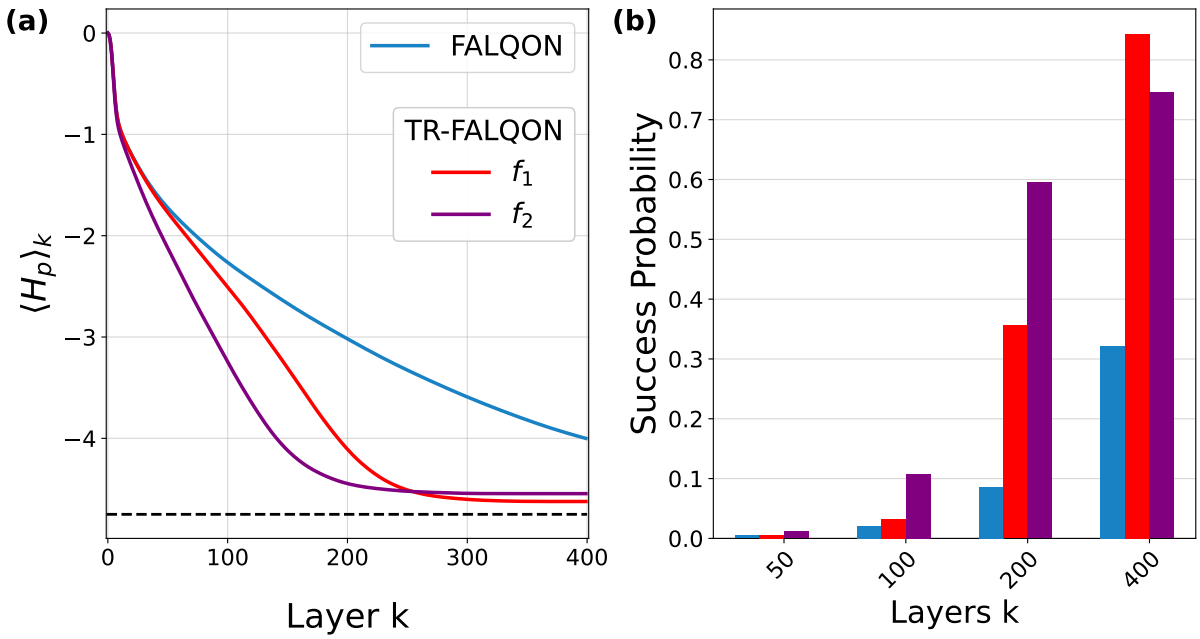

Figura 17 – Comparação do desempenho do FALQON e TR-FALQON para diferentes funções de reescalonamento temporal. O painel (a) exibe a convergência da função objetivo $\langle \psi_k | H | \psi_k \rangle$ em função do número de camadas k, com a linha tracejada indicando a energia do estado fundamental. O painel (b) apresenta as probabilidades de obter uma solução do problema em certas camadas.

Eq. (134). O TR-FALQON foi simulado com $a = 2$ e $t_f = 120$. Observa-se que, para ambas as funções, o TR-FALQON apresenta uma queda inicial mais acentuada no valor esperado da energia $\langle H_p \rangle$, sendo essa queda ligeiramente mais pronunciada com a função f_2. Contudo, após essa redução inicial, a energia tende a se estabilizar, com a simulação usando f_2 estagnando em um valor um pouco mais alto do que aquele obtido com f_1.

Em contraste, o FALQON exibe uma redução mais lenta e contínua da energia, sem apresentar estagnação visível no intervalo considerado. Ainda assim, mesmo com essa limitação, o TR-FALQON conseguiu evidenciar soluções corretas com menos camadas. Com 200 camadas, a probabilidade de sucesso com a função f_2 se aproximou de 60%, enquanto com f_1 superou 30%. No mesmo número de camadas, o FALQON permaneceu abaixo de 10%. Com 400 camadas, o desempenho do TR-FALQON com f_1 superou f_2, alcançando uma probabilidade superior a 80%, enquanto f_2 manteve desempenho elevado, com mais de 70%. O FALQON, por sua vez, obteve probabilidade em torno de 30%. Esses resultados indicam que o TR-FALQON pode acelerar a convergência para soluções de interesse, destacando-se em regimes de baixa profundidade.

Em seguida, investigou-se o impacto da variação dos parâmetros a e t_f no desempenho do TR-FALQON ao manter $\Delta\tau$ fixo em $0,04$, utilizando a função de reescalonamento f_1.

A Figura 20 mostra a influência do parâmetro a no desempenho do TR-FALQON,

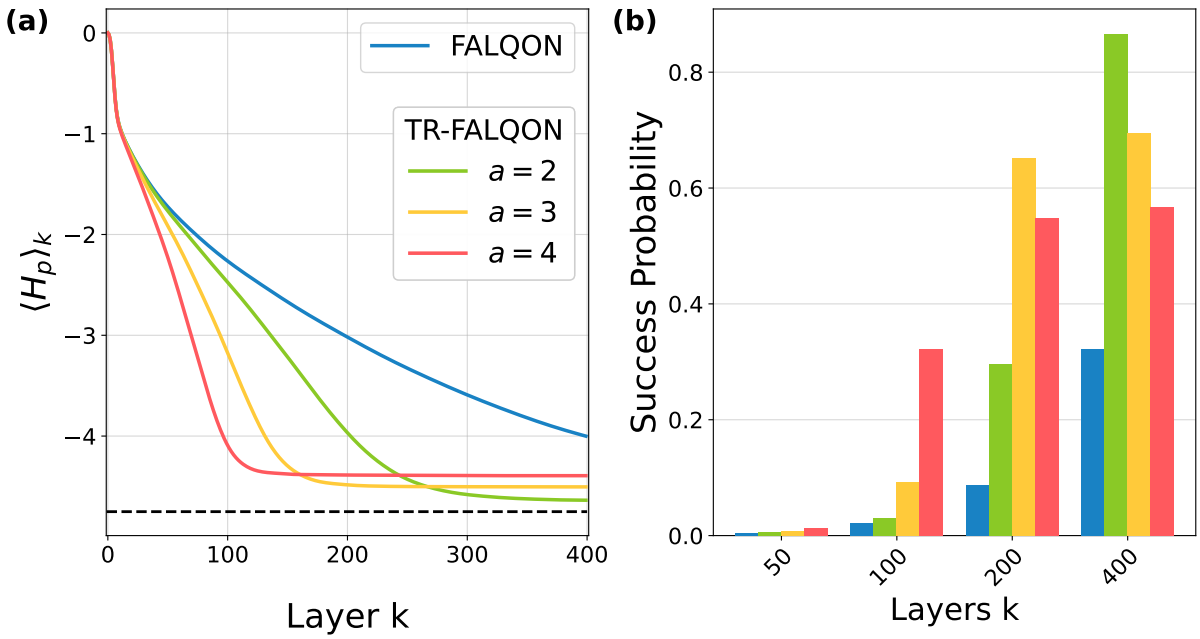

Figura 18 – Comparação do desempenho do FALQON e TR-FALQON para diferentes valores do parâmetro de controle de contração temporal a. O painel (a) exibe a convergência da função objetivo $\langle\psi_k|H|\psi_k\rangle$ em função do número de camadas k, com a linha tracejada indicando a energia do estado fundamental. O painel (b) apresenta as probabilidades de obter uma solução do problema em certas camadas.

comparando diferentes valores de a com o FALQON padrão, com $t_f = 120$. No TR-FALQON, observa-se que valores maiores de a aceleram a queda inicial do valor esperado da energia, promovendo uma convergência mais rápida nas camadas iniciais. No entanto, esse ganho inicial vem acompanhado de uma estagnação prematura, em que o algoritmo atinge um patamar intermediário e não avança significativamente em direção ao estado fundamental nas camadas seguintes.

Além disso, a estagnação ocorre em valores mais altos de energia para valores maiores de a, o que resulta em menores probabilidades de medir os estados que codificam a solução do problema. Já o FALQON exibe uma redução mais lenta, porém constante, exigindo mais camadas para atingir probabilidades comparáveis.

Para $a = 2$, o TR-FALQON teve uma queda inicial menos acentuada que para $a = 3$ e $a = 4$, mas superou ambos em desempenho final. Com 400 camadas, a probabilidade de obter a solução ultrapassou 80%. Já para $a = 4$, observou-se uma rápida redução da energia nas primeiras 100 camadas, seguida de uma estagnação; a probabilidade estabilizou em torno de 50% nas camadas 200 e 400, sem ganho adicional.

Esses resultados indicam que, mantendo $\Delta\tau$ fixo, valores maiores de a aceleram a convergência inicial, mas também aumentam a tendência de estagnação. Um valor intermediário de a pode oferecer melhor equilíbrio entre velocidade de convergência e fidelidade da solução.

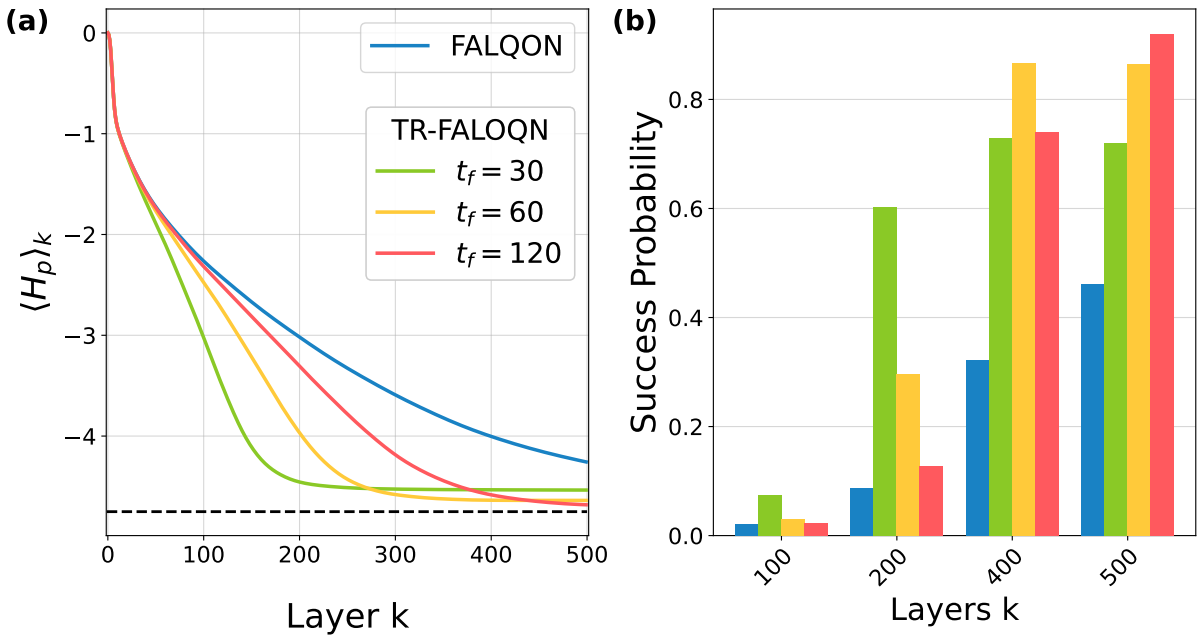

Figura 19 – Comparação do desempenho do FALQON e TR-FALQON para diferentes valores do tempo final t_f. O painel (a) exibe a convergência da função objetivo $\langle \psi_k | H | \psi_k \rangle$ em função do número de camadas k, com a linha tracejada indicando a energia do estado fundamental. O painel (b) apresenta as probabilidades de obter uma solução do problema em certas camadas.

A Figura 19 apresenta uma comparação entre o FALQON e o TR-FALQON para diferentes valores do tempo final t_f, com o parâmetro de contração temporal fixado em $a = 2$. Os resultados mostram que o desempenho do TR-FALQON é sensivelmente afetado pela escolha de t_f. Valores menores de t_f promovem uma queda mais rápida do valor esperado da energia nas camadas iniciais, porém também induzem uma estagnação precoce. Por outro lado, valores maiores de t_f resultam em uma queda inicial mais lenta, mas permitem uma progressão mais contínua em direção ao estado fundamental, evitando platôs prematuros.

Para $t_f = 30$, observou-se uma queda acentuada até aproximadamente 200 camadas, seguida de estagnação. A probabilidade de obter a solução do problema era cerca de 60% nessa região, estabilizando em torno de 70% nas camadas 400 e 500. Com $t_f = 60$, a estagnação ocorreu mais tardiamente, por volta da camada 300, mas com maior fidelidade: a probabilidade ultrapassou 80%. Já para $t_f = 120$, o decaimento foi mais lento nas camadas iniciais, porém a curva ultrapassou as demais por volta da camada 500, atingindo maior fidelidade que os outros dois casos.

Esses resultados indicam que o parâmetro t_f controla o ritmo de evolução do algoritmo: valores menores favorecem convergência rápida, mas podem limitar a aproximação ao estado fundamental devido à estagnação prematura. Em contrapartida, tempos finais mais longos permitem uma convergência mais lenta, porém mais eficaz em profundidades maiores.

Por fim, analisou-se a capacidade do TR-FQA de preparar o estado fundamental no modelo ANNNI, considerando diferentes configurações de cadeia L e parâmetros κ e g do Hamiltoniano.

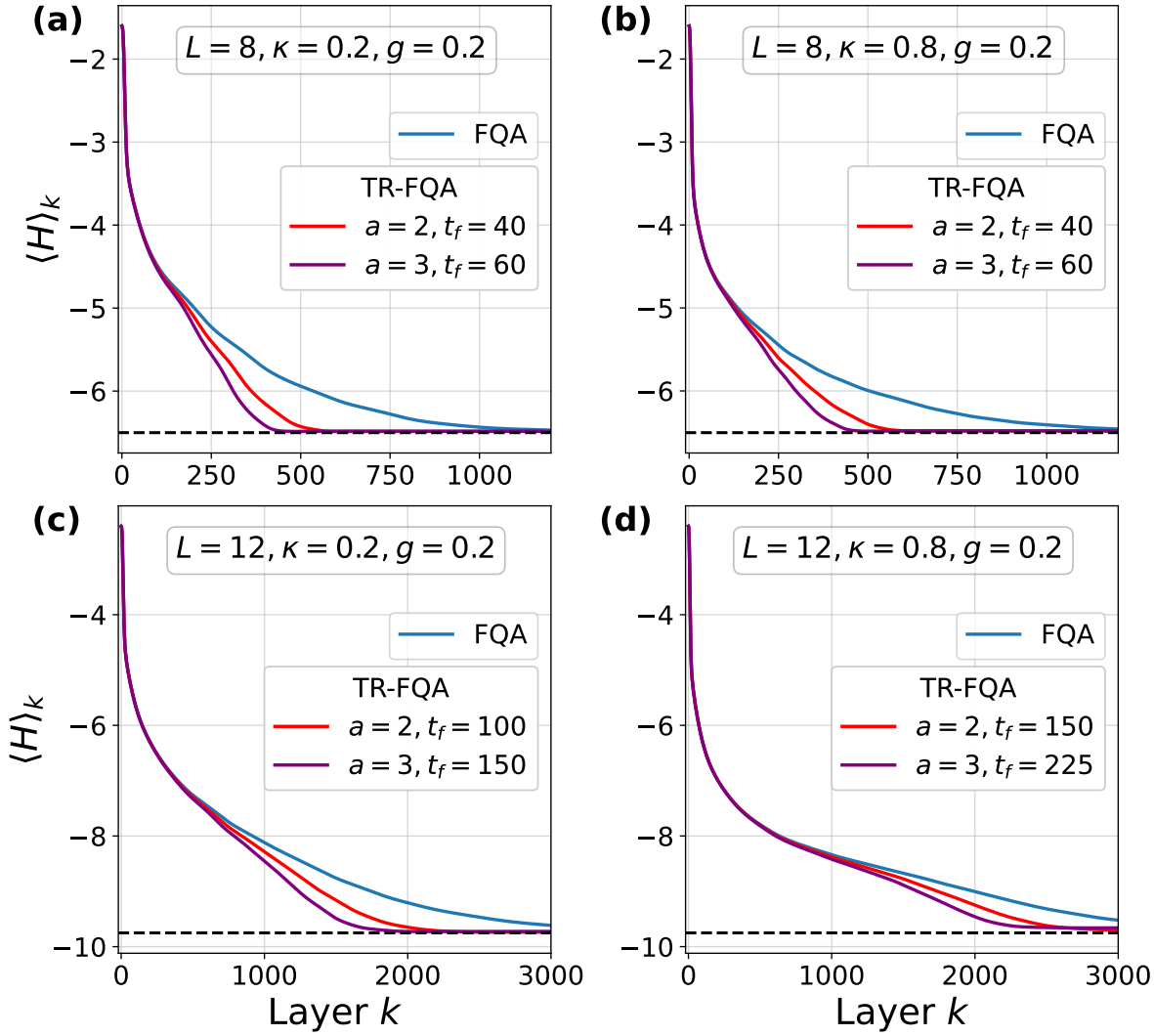

Figura 20 – Simulações numéricas comparando FQA e TR-FQA aplicados ao modelo ANNNI, para diferentes configurações de cadeia e valores de κ e g. As simulações com TR-FQA utilizaram a função de reescalonamento f_1, com os valores de a e t_f indicados em cada painel. Para cadeias com $L = 8$ sítios, adotou-se $\Delta t = \Delta\tau = 0.01$, enquanto para $L = 12$ sítios, usou-se $\Delta t = \Delta\tau = 0.005$. As linhas tracejadas indicam a energia do estado fundamental, enquanto as curvas mostram a convergência da função de custo $J = \langle \Psi_k | H | \Psi_k \rangle$ em função do número de camadas k.

A Figura 20 apresenta os resultados da aplicação dos algoritmos FQA e TR-FQA ao modelo ANNNI, considerando cadeias de tamanhos distintos. Os dados indicam que, à medida que o número de sítios aumenta, são necessárias mais camadas para atingir a convergência. Ainda assim, o TR-FQA apresenta desempenho superior em relação ao FQA, alcançando valores

mais próximos da energia do estado fundamental com menos camadas.

Dentre as configurações testadas, o protocolo com $a = 3$ apresentou melhor desempenho, reduzindo a profundidade necessária do circuito em mais de 500 camadas em comparação com o FQA padrão. Os resultados também sugerem que, para manter a eficiência da convergência em sistemas maiores ou com valores mais altos de a, é necessário ajustar o tempo final t_f adequadamente.

11 Conclusão

Neste trabalho, foi investigado o algoritmo de Feedback Quântico (FQA) em diferentes contextos, incluindo a preparação de estados físicos e problemas de otimização combinatória. A análise concentrou-se na convergência do algoritmo e na proposição de estratégias para aprimorar seu desempenho. As contribuições foram organizadas em três frentes: o papel das simetrias do Hamiltoniano, a redução da profundidade dos circuitos via agrupamento de camadas e o uso de reescalonamento temporal.

Na primeira frente, analisou-se o impacto das simetrias do Hamiltoniano sobre a evolução do FQA. Verificou-se que o algoritmo preserva as simetrias do estado inicial, impedindo a transição entre diferentes setores do espaço de Hilbert. Isso implica que a escolha do estado inicial é determinante para a preparação do estado alvo. Em particular, superposições entre vetores de diferentes setores evoluem de forma independente, sem acoplamento entre os blocos. A estrutura de blocos também foi explorada como recurso, permitindo aplicar o WFQAE em setores não degenerados para extrair autovalores de forma controlada e contornar limitações associadas à degenerescência do espectro.

Na segunda frente, foi proposta a estratégia LGA-FQA para reduzir a profundidade dos circuitos, por meio do agrupamento de camadas. As versões LGA-FQA-1 e LGA-FQA-2 apresentaram comportamentos distintos. A primeira demonstrou convergência inicial mais rápida, porém com maior tendência à estagnação em regimes de alto número de camadas agrupadas ou grandes incrementos temporais. Já a segunda versão apresentou desempenho mais estável e consistente em profundidades elevadas, embora com custo maior de implementação, dado o aumento efetivo na profundidade das camadas agrupadas.

Na terceira frente, foi introduzido o algoritmo TR-FQA, baseado no reescalonamento temporal da dinâmica de feedback, incorporado à função de controle do FQA para modificar sua evolução ao longo do tempo. Os resultados indicaram que o TR-FQA apresentou uma convergência mais rápida que o FQA convencional, especialmente nas iterações iniciais. No entanto, observou-se estagnação da convergência em alguns casos, principalmente para valores pequenos de t_f e altos de a, sendo esse efeito mais recorrente com a função de reescalonamento f_2 do que com f_1. Ainda assim, o TR-FALQON demonstrou um desempenho satisfatório em relação ao FALQON.

Aplicadas ao problema MaxCut, as abordagens LGA-FALQON e TR-FALQON demonstraram potencial na evidência da solução com baixa profundidade de circuito, mesmo sem necessariamente atingir o estado fundamental. Já no caso do modelo ANNNI, as estratégias propostas permitiram a preparação do estado fundamental com maior eficiência. Esses resultados reforçam a utilidade dos métodos tanto para problemas de otimização combinatória quanto

para a simulação de sistemas físicos.

Em síntese, os resultados obtidos indicam que o FQA pode ser aprimorado por meio da exploração das simetrias do sistema, do agrupamento de camadas e da adaptação da dinâmica temporal. Essas estratégias contribuem para mitigar limitações relacionadas à profundidade dos circuitos e à degenerescência espectral, ampliando o escopo de aplicação do algoritmo. No entanto, mesmo com as melhorias propostas, o FQA e suas variantes ainda demandam circuitos com profundidades superiores às atualmente suportadas por dispositivos quânticos da era NISQ, limitando sua aplicação prática em hardware real. Ainda assim, os resultados fornecem um direcionamento relevante para o desenvolvimento de versões mais eficientes e, futuramente, compatíveis com a execução em plataformas ruidosas.

Referências

1 LARSEN, J. B. et al. Feedback-based quantum algorithms for ground state preparation. *Phys. Rev. Res.*, American Physical Society, v. 6, p. 033336, Sep 2024. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevResearch.6.033336>.

2 RAHMAN, S. A.; KARABACAK Özkan; WISNIEWSKI, R. *Weighted Feedback-Based Quantum Algorithm for Excited States Calculation*. 2024. Disponível em: <https://arxiv.org/abs/2404.19386>.

3 SUZUKI, S.; INOUE, J. ichi; CHAKRABARTI, B. Quantum ising phases and transitions in transverse ising models. *Springer*, Berlin Heidelberg, 2013.

4 CHANDRA, A. K.; DASGUPTA, S. Floating phase in the one-dimensional transverse axial next-nearest-neighbor ising model. *Phys. Rev. E*, American Physical Society, v. 75, p. 021105, Feb 2007. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevE.75.021105>.

5 MATTIS, D. C. *The Theory of Magnetism II: Thermodynamics and Statistical Mechanics*. Berlin, Heidelberg: Springer-Verlag, 1985. II. (Springer Series in Solid-State Sciences, II). Section 3.6.

6 DEUTSCH, D.; JOZSA, R. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, v. 439, n. 1907, p. 553–558, 1992. Disponível em: <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1992.0167>.

7 GROVER, L. K. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, American Physical Society, v. 79, p. 325–328, Jul 1997. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevLett.79.325>.

8 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, ago. 2018. ISSN 2521-327X. Disponível em: <https://doi.org/10.22331/q-2018-08-06-79>.

9 BHARTI, K. et al. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.*, American Physical Society, v. 94, p. 015004, Feb 2022. Disponível em: <https://link.aps.org/doi/10.1103/RevModPhys.94.015004>.

10 MCCLEAN, J. R. et al. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, IOP Publishing, v. 18, n. 2, p. 023023, fev. 2016. ISSN 1367-2630. Disponível em: <http://dx.doi.org/10.1088/1367-2630/18/2/023023>.

11 CEREZO, M. et al. Variational quantum algorithms. *Nature Reviews Physics*, Springer Science and Business Media LLC, v. 3, n. 9, p. 625–644, ago. 2021. ISSN 2522-5820. Disponível em: <http://dx.doi.org/10.1038/s42254-021-00348-9>.

12 PERUZZO, A. et al. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, Springer Science and Business Media LLC, v. 5, n. 1, jul 2014. Disponível em: <https://doi.org/10.1038%2Fncomms5213>.

13 FEDOROV, D. A. et al. *VQE Method: A Short Survey and Recent Developments*. 2021. Disponível em: <https://arxiv.org/abs/2103.08505>.

14 DÍEZ-VALLE, P.; PORRAS, D.; GARCÍA-RIPOLL, J. J. Quantum variational optimization: The role of entanglement and problem hardness. *Phys. Rev. A*, American Physical Society, v. 104, p. 062426, Dec 2021. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevA.104.062426>.

15 TILLY, J. et al. The variational quantum eigensolver: A review of methods and best practices. *Physics Reports*, Elsevier BV, v. 986, p. 1–128, nov. 2022. ISSN 0370-1573. Disponível em: <http://dx.doi.org/10.1016/j.physrep.2022.08.003>.

16 LE, T. V.; KEKATOS, V. Solving constrained optimization problems via the variational quantum eigensolver with constraints. *Phys. Rev. A*, American Physical Society, v. 110, p. 022430, Aug 2024. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevA.110.022430>.

17 NAKANISHI, K. M.; MITARAI, K.; FUJII, K. Subspace-search variational quantum eigensolver for excited states. *Phys. Rev. Res.*, American Physical Society, v. 1, p. 033062, Oct 2019. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevResearch.1.033062>.

18 HIGGOTT, O.; WANG, D.; BRIERLEY, S. Variational Quantum Computation of Excited States. *Quantum*, Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, v. 3, p. 156, jul. 2019. ISSN 2521-327X. Disponível em: <https://doi.org/10.22331/q-2019-07-01-156>.

19 FARHI, E.; GOLDSTONE, J.; GUTMANN, S. *A Quantum Approximate Optimization Algorithm*. 2014. Disponível em: <https://arxiv.org/abs/1411.4028>.

20 ICHIKAWA, K. et al. *Optimal elemental configuration search in crystal using quantum approximate optimization algorithm*. 2025. Disponível em: <https://arxiv.org/abs/2503.09356>.

21 JIANG, X. et al. Qaoa-based mrmr algorithm for feature selection. In: *Proceedings of the 2023 International Conference on Advances in Artificial Intelligence and Applications*. New York, NY, USA: Association for Computing Machinery, 2024. (AAIA '23), p. 277–282. ISBN 9798400708268. Disponível em: <https://doi.org/10.1145/3603273.3631193>.

22 BECHTOLD, M. et al. Investigating the effect of circuit cutting in qaoa for the maxcut problem on nisq devices. *Quantum Science and Technology*, IOP Publishing, v. 8, n. 4, p. 045022, set. 2023. ISSN 2058-9565. Disponível em: <http://dx.doi.org/10.1088/2058-9565/acf59c>.

23 BITTEL, L.; KLIESCH, M. Training variational quantum algorithms is np-hard. *Phys. Rev. Lett.*, American Physical Society, v. 127, p. 120502, Sep 2021. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevLett.127.120502>.

24 MAGANN, A. B. et al. Feedback-based quantum optimization. *Physical Review Letters*, American Physical Society (APS), v. 129, n. 25, dec 2022. Disponível em: <https://doi.org/10.1103%2Fphysrevlett.129.250502>.

25 MAGANN, A. B. et al. Lyapunov-control-inspired strategies for quantum combinatorial optimization. *Phys. Rev. A*, American Physical Society, v. 106, p. 062414, Dec 2022. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevA.106.062414>.

26 GRIVOPOULOS, S.; BAMIEH, B. Lyapunov-based control of quantum systems. In: . [S.l.: s.n.], 2004. v. 1, p. 434 – 438 Vol.1. ISBN 0-7803-7924-1.

27 D'ALESSANDRO, D. *Introduction to Quantum Control and Dynamics*. 2. ed. CRC Press, 2021. 253–263 p. (Advances in Applied Mathematics). ISBN 9781000394993. Disponível em: <https://books.google.com.br/books?id=n6A4EAAAQBAJ>.

28 CONG, S.; MENG, F. A survey of quantum lyapunov control methods. *The Scientific World Journal*, Wiley, v. 2013, n. 1, jan. 2013. ISSN 1537-744X. Disponível em: <http://dx.doi.org/10.1155/2013/967529>.

29 MAGANN, A. B.; ECONOMOU, S. E.; ARENZ, C. Randomized adaptive quantum state preparation. *Phys. Rev. Res.*, American Physical Society, v. 5, p. 033227, Sep 2023. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevResearch.5.033227>.

30 ARAI, D. et al. *Scalable circuit depth reduction in feedback-based quantum optimization with a quadratic approximation*. 2024. Disponível em: <https://arxiv.org/abs/2407.17810>.

31 RAHMAN, S. A.; KARABACAK Özkan; WISNIEWSKI, R. *Feedback-Based Quantum Algorithm for Excited States Calculation*. 2024. Disponível em: <https://arxiv.org/abs/2404.04620>.

32 LEWIS, M.; GLOVER, F. Quadratic unconstrained binary optimization problem preprocessing: Theory and empirical analysis. *Networks*, v. 70, n. 2, p. 79–97, 2017. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/net.21751>.

33 GLOVER, F. W.; KOCHENBERGER, G. A. A tutorial on formulating QUBO models. *CoRR*, abs/1811.11538, 2018. Disponível em: <http://arxiv.org/abs/1811.11538>.

34 LEE, H.; JUN, K. Range dependent hamiltonian algorithms for numerical QUBO formulation. *Scientific Reports*, v. 15, n. 1, p. 8819, mar. 2025.

35 IRIE, H. et al. Quantum annealing of vehicle routing problem with time, state and capacity. In: FELD, S.; LINNHOFF-POPIEN, C. (Ed.). *Quantum Technology and Optimization Problems*. Cham: Springer International Publishing, 2019. p. 145–156. ISBN 978-3-030-14082-3.

36 OSTERMANN, B. et al. *Benchmarking of quantum and classical SDP relaxations for QUBO formulations of real-world logistics problems*. 2025. Disponível em: <https://arxiv.org/abs/2503.10801>.

37 PHILLIPSON, F. *Quantum Computing in Logistics and Supply Chain Management an Overview*. 2025. Disponível em: <https://arxiv.org/abs/2402.17520>.

38 EWALD, D. Application of quantum computers and their unique properties for constrained optimization in engineering problems: Welded beam design. *Electronics*, v. 14, n. 5, 2025. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/14/5/1027>.

39 DATE, P.; ARTHUR, D.; PUSEY-NAZZARO, L. Qubo formulations for training machine learning models. *Scientific Reports*, Springer Science and Business Media LLC, v. 11, n. 1, maio 2021. ISSN 2045-2322. Disponível em: <http://dx.doi.org/10.1038/s41598-021-89461-4>.

40 HELLSTERN, G.; DEHN, V.; ZAEFFERER, M. Quantum computer based feature selection in machine learning. *IET Quantum Communication*, v. 5, n. 3, p. 232–252, 2024. Disponível em: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/qtc2.12086>.

41 LUCAS, A. Ising formulations of many np problems. *Frontiers in Physics*, v. 2, 2014. ISSN 2296-424X. Disponível em: <https://www.frontiersin.org/journals/physics/articles/10.3389/fphy.2014.00005>.

42 LUCAS, A. Ising formulations of many np problems. *Frontiers in Physics*, Frontiers Media SA, v. 2, 2014. ISSN 2296-424X. Disponível em: <http://dx.doi.org/10.3389/fphy.2014.00005>.

43 TINKHAM, M. *Group Theory and Quantum Mechanics*. Dover Publications, 2012. (Dover Books on Chemistry). ISBN 9780486131665. Disponível em: <https://books.google.com.br/books?id=oe7DAgAAQBAJ>.

44 LANDSMAN, K. *Foundations of Quantum Theory, From Classical Concepts to Operator Algebras*. Springer Nature, 2017. 125–188 p. Disponível em: <https://doi.org/10.1007/978-3-319-51777-3>.

45 LIN, Y.-H.; SHAO, S.-H. $\mathbb{Z}_N$ symmetries, anomalies, and the modular bootstrap. *Phys. Rev. D*, American Physical Society, v. 103, p. 125001, Jun 2021. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevD.103.125001>.

46 BELLAC, M. L. *Quantum Physics*. [S.l.]: Cambridge University Press, 2006. 125–132 p.

47 JUNG, J.-H.; NOH, J. D. Guide to exact diagonalization study of quantum thermalization. *Journal of the Korean Physical Society*, Korean Physical Society, v. 76, n. 8, p. 670–683, abr. 2020. ISSN 1976-8524. Disponível em: <http://dx.doi.org/10.3938/jkps.76.670>.

48 MALVEZZI, A. L. An introduction to numerical methods in low-dimensional quantum systems. *Brazilian journal of physics*, v. 33, p. 55–72, 2003.

49 SELKE, W. The annni model — theoretical analysis and experimental application. *Physics Reports*, v. 170, n. 4, p. 213–264, 1988. ISSN 0370-1573. Disponível em: <https://www.sciencedirect.com/science/article/pii/0370157388901408>.

50 CANABARRO, A. et al. Unveiling phase transitions with machine learning. *Phys. Rev. B*, American Physical Society, v. 100, p. 045129, Jul 2019. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevB.100.045129>.

51 YANG, Y.-T.; LUO, H.-G. *Pattern description of the ground state properties of the one-dimensional axial next-nearest-neighbor Ising model in a transverse field*. 2023.

52 FERREIRA-MARTINS, A. J. et al. Detecting quantum phase transitions in a frustrated spin chain via transfer learning of a quantum classifier algorithm. *Phys. Rev. A*, American Physical Society, v. 109, p. 052623, May 2024. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevA.109.052623>.

53 RIEGER, H.; UIMIN, G. The one-dimensional annni model in a transverse field: analytic and numerical study of effective hamiltonians. *Zeitschrift für Physik B Condensed Matter*, Springer Science and Business Media LLC, v. 101, n. 4, p. 597–611, dez. 1996. ISSN 1431-584X. Disponível em: <http://dx.doi.org/10.1007/s002570050252>.

54 CEA, M. et al. *Exploring the Phase Diagram of the quantum one-dimensional ANNNI model*. 2024.

55 RATTIGHIERI, L. A. M. et al. *Accelerating Feedback-Based Quantum Algorithms through Time Rescaling*. 2025. Disponível em: <https://arxiv.org/abs/2504.01256>.

56 BERNARDO, B. d. L. Time-rescaled quantum dynamics as a shortcut to adiabaticity. *Phys. Rev. Res.*, American Physical Society, v. 2, p. 013133, Feb 2020. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevResearch.2.013133>.

57 FERREIRA, J. L. M. et al. *Shortcuts to adiabaticity designed via time-rescaling follow the same transitionless route*. 2024. Disponível em: <https://arxiv.org/abs/2406.07433>.

58 PEXE, G. E. L. et al. Using a feedback-based quantum algorithm to analyze the critical properties of the annni model without classical optimization. *Phys. Rev. B*, American Physical Society, v. 110, p. 224422, Dec 2024. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevB.110.224422>.

59 NIELSEN, M. A.; CHUANG, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. [S.l.]: Cambridge University Press, 2010.

60 SUZUKI, M. Generalized trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Communications in Mathematical Physics*, v. 51, n. 2, p. 183–190, 1976. ISSN 1432-0916. Disponível em: <https://doi.org/10.1007/BF01609348>.

61 ICHINOSE, T.; TAMURA, H. Error estimate in operator norm for trotter-kato product formula. *Integral Equations and Operator Theory*, v. 27, n. 2, p. 195–207, 1997. ISSN 1420-8989. Disponível em: <https://doi.org/10.1007/BF01191532>.

62 HALL, B. C. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. 2. ed. Springer, 2015. v. 222. 109–119 p. (Graduate Texts in Mathematics, v. 222). ISBN 978-3-319-13466-6. Disponível em: <https://doi.org/10.1007/978-3-319-13467-3>.

63 ACHILLES, R.; BONFIGLIOLI, A. The early proofs of the theorem of campbell, baker, hausdorff, and dynkin. *Archive for History of Exact Sciences*, v. 66, n. 3, p. 295–358, 2012. ISSN 1432-0657. Disponível em: <https://doi.org/10.1007/s00407-012-0095-8>.

Apêndices

APÊNDICE A – Fórmula de Trotter

Considere a expansão de Taylor da exponencial de um operador A:

$$e^{At} = \sum_{p=0}^{\infty} \frac{t^p}{p!} A^p = \mathbb{1} + tA + \frac{t^2}{2!}A^2 + \frac{t^3}{3!}A^3 + \dots \tag{149}$$

Nos interessa estudar a forma da exponencial de uma soma de dois operadores hermitianos A e B, ou seja:

$$e^{(A+B)t}$$

Primeiramente, analisamos o caso em que os operadores comutam, ou seja, $[A, B] = 0$. Nesse cenário, vale a relação:

$$e^{At}e^{Bt} = \left(\sum_{p=0}^{\infty} \frac{t^p}{p!} A^p\right)\left(\sum_{q=0}^{\infty} \frac{t^q}{q!} B^q\right) \tag{150}$$

Como $A^p B^q = B^q A^p$, podemos aplicar o produto de Cauchy para séries:

$$\left(\sum_{p=0}^{\infty} a_p\right)\left(\sum_{q=0}^{\infty} b_q\right) = \sum_{p=0}^{\infty}\sum_{k=0}^{n} a_k b_{p-k} \tag{151}$$

Aplicando essa fórmula:

$$\begin{aligned} e^{At}e^{Bt} &= \sum_{p=0}^{\infty}\sum_{k=0}^{p} \frac{(A^k t^k)}{k!}\frac{(B^{p-k}t^{p-k})}{(p-k)!} \\ &= \sum_{p=0}^{\infty} \frac{t^p}{p!}\sum_{k=0}^{p} \frac{p!}{k!(p-k)!} A^k B^{p-k} \end{aligned} \tag{152}$$

Note que o termo dentro da soma corresponde à expansão binomial de $(A+B)^p$, ou seja:

$$(A+B)^p = \sum_{k=0}^{p} \binom{p}{k} A^k B^{p-k} = \sum_{k=0}^{p} \frac{p!}{k!(p-k)!} A^k B^{p-k} \tag{153}$$

Portanto:

$$\begin{aligned} e^{At}e^{Bt} &= \sum_{p=0}^{\infty} \frac{t^p}{p!}(A+B)^p \\ &= e^{(A+B)t} \end{aligned} \tag{154}$$

Ou seja, quando A e B comutam, a exponencial da soma é igual ao produto das exponenciais.

Nos casos em que A e B não comutam, ou seja, $[A, B] \neq 0$, a Eq. (154) deixa de ser válida. Nesses cenários, recorremos fórmula de Trotter (59, 60, 61), que permite expressar a exponencial da soma de dois operadores não comutantes como o limite de um produto de exponenciais individuais:

$$\lim_{n\to\infty} \left(e^{At/n} e^{Bt/n}\right)^n = e^{(A+B)t} \tag{155}$$

Para demonstrar essa relação, consideramos inicialmente a expansão de Taylor até a ordem $1/n$ dos termos $e^{At/n}$ e $e^{Bt/n}$. Como t/n é pequeno, temos:

$$e^{At/n} = \mathbb{1} + \frac{t}{n} A + \mathcal{O}\left(\frac{1}{n^2}\right) \tag{156}$$

Multiplicando as duas exponenciais:

$$e^{At/n} e^{Bt/n} = \mathbb{1} + \frac{t}{n}(A + B) + \mathcal{O}\left(\frac{1}{n^2}\right) \tag{157}$$

Agora, elevamos a equação (157) à potência n:

$$\left(e^{At/n} e^{Bt/n}\right)^n = \left(\mathbb{1} + \frac{t}{n}(A + B) + \mathcal{O}\left(\frac{1}{n^2}\right)\right)^n \tag{158}$$

Expandindo esse termo como uma série binomial, obtemos:

$$\left(e^{At/n} e^{Bt/n}\right)^n = \mathbb{1} + \sum_{k=1}^{n} \binom{n}{k} \frac{1}{n^k} [t(A + B)]^k + \mathcal{O}\left(\frac{1}{n^2}\right) \tag{159}$$

Para simplificar o termo binomial, analisamos:

$$\begin{aligned}
\binom{n}{k} \frac{1}{n^k} &= \frac{n!}{k!(n-k)!} \frac{1}{n^k} \\
&= \frac{1}{k!} \cdot \frac{n(n-1)\cdots(n-k+1)}{n^k} \\
&= \frac{1}{k!} \left(1 - \frac{1}{n}\right) \left(1 - \frac{2}{n}\right) \cdots \left(1 - \frac{k-1}{n}\right) \\
&= \frac{1}{k!} \left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right)
\end{aligned} \tag{160}$$

Substituindo isso de volta na expansão:

$$\left(e^{At/n} e^{Bt/n}\right)^n = \mathbb{1} + \sum_{k=1}^{n} \frac{1}{k!} [t(A + B)]^k \left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right) + \mathcal{O}\left(\frac{1}{n^2}\right) \tag{161}$$

Tomando o limite de $n \to \infty$, os termos de ordem $1/n$ e $1/n^2$ desaparecem, e obtemos:

$$\begin{aligned}
\lim_{n\to\infty}\left(e^{At/n}e^{Bt/n}\right)^n &= \mathbb{1} + \sum_{k=1}^{n}\frac{1}{k!}[t(A+B)]^k \lim_{n\to\infty}\left(1+\mathcal{O}\left(\frac{1}{n}\right)\right) + \lim_{n\to\infty}\mathcal{O}\left(\frac{1}{n^2}\right) \\
&= \mathbb{1} + \sum_{k=1}^{n}\frac{1}{k!}[t(A+B)]^k \\
&= e^{(A+B)t}
\end{aligned} \tag{162}$$

Ou seja, recuperamos a exponencial da soma, mesmo no caso em que os operadores não comutam, através de um limite apropriado.

Essa fórmula é a base da aproximação de Trotter de primeira ordem, que pode ser escrita como:

$$e^{(A+B)\delta t} \approx e^{A\Delta t}e^{B\Delta t} + \mathcal{O}\left(\Delta t^3\right) \tag{163}$$

onde $\Delta t = t/n$ é pequeno. Para valores finitos de n, esse produto não é exatamente igual à exponencial da soma, mas fornece uma boa aproximação quando Δt é suficientemente pequeno.

Além disso, uma aproximação mais precisa pode ser obtida simetricamente, utilizando a chamada aproximação de segunda ordem (Suzuki):

$$e^{(A+B)\Delta t} \approx e^{A\Delta t/2}e^{B\Delta t}e^{A\Delta t/2} + \mathcal{O}\left(\Delta t^3\right) \tag{164}$$

APÊNDICE B – Identidade de Baker-Campbell-Hausdorff (BCH)

A Identidade de Baker-Campbell-Hausdorff (BCH) permite expressar o logaritmo do produto de duas exponenciais de operadores A e B como uma única exponencial (62, 63):

$$\ln\left(e^A e^B\right) = Z \tag{165}$$

ou, de forma equivalente:

$$e^A e^B = e^Z \tag{166}$$

O operador Z é dado por uma série que envolve comutadores entre A e B:

$$Z = A + B + \frac{1}{2}[A, B] + \frac{1}{12}[A, [A, B]] + \frac{1}{12}[B, [B, A]] + \cdots \tag{167}$$

Para obter essa forma explícita de Z, é necessário analisar a ação de operadores conjugados, em particular $e^A B e^{-A}$. Para isso, define-se a função:

$$\tilde{B}(t) = e^{At} B e^{-At} \tag{168}$$

Derivando em relação a t:

$$\begin{aligned} \frac{d\tilde{B}}{dt}(t) &= \frac{d}{dt}\left(e^{At} B e^{-At}\right) \\ &= e^{At} A B e^{-At} - e^{At} B A e^{-At} \\ &= e^{At}[A, B]e^{-At} = [A, \tilde{B}(t)] \end{aligned} \tag{169}$$

Derivando uma segunda vez:

$$\frac{d^2\tilde{B}}{dt^2}(t) = \frac{d}{dt}\left[A, \tilde{B}(t)\right] = \left[A, \frac{d\tilde{B}}{dt}(t)\right] = [A, [A, \tilde{B}(t)]] \tag{170}$$

De forma recursiva:

$$\frac{d^n\tilde{B}}{dt^n}(t) = \left[A, \frac{d^{n-1}\tilde{B}}{dt^{n-1}}(t)\right] \tag{171}$$

Esse comportamento pode ser descrito pela aplicação iterada do operador adjunto ad_A, definido por:

$$\mathsf{ad}_A(B) = [A, B] \tag{172}$$

Logo, a derivada de ordem n de $\tilde{B}(t)$ é:

$$\frac{d^n \tilde{B}}{dt^n}(t) = (\mathsf{ad}_A)^n(B) \tag{173}$$

Expandindo $\tilde{B}(t)$ em série de Taylor ao redor de $t = 0$, com $\tilde{B}(0) = B$:

$$\tilde{B}(t) = B + \sum_{n=1}^{\infty} \frac{t^n}{n!}(\mathsf{ad}_A)^n(B) \tag{174}$$

Para $t = 1$, obtém-se:

$$e^A B e^{-A} = B + \sum_{n=1}^{\infty} \frac{1}{n!}(\mathsf{ad}_A)^n(B) \tag{175}$$

De modo análogo, pode-se demonstrar que:

$$e^{-A} B e^{A} = B + \sum_{n=1}^{\infty} \frac{(-1)^n}{n!}(\mathsf{ad}_A)^n(B) \tag{176}$$

Para calcular a derivada de $e^{A(t)}$ em relação ao tempo, começamos pela definição da série de Taylor da exponencial:

$$\frac{d}{dt}e^{A(t)} = \frac{d}{dt}\left(1 + A + \frac{1}{2}A^2 + \frac{1}{3!}A^3 + \dots\right) \tag{177}$$

Aplicando a regra do produto para derivadas de potências de $A(t)$, obtemos:

$$\frac{d}{dt}e^{A(t)} = A' + \frac{1}{2}(A'A + AA') + \frac{1}{3!}(A'AA + AA'A + AAA') + \dots \tag{178}$$

Essa expressão pode ser reescrita de forma compacta como:

$$\frac{d}{dt}e^{A(t)} = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} \frac{1}{(n+m+1)!}A^n A' A^m \tag{179}$$

com $A' = \frac{dA}{dt}$.

Por outro lado, consideramos a seguinte identidade envolvendo integrais:

$$\int_0^1 ds\; e^{(1-s)A}\frac{dA}{dt}e^{sA} \tag{180}$$

Expandindo as exponenciais em séries de potências:

$$\int_0^1 ds \left(\sum_{n=0}^{\infty} \frac{(1-s)^n A^n}{n!} \right) A' \left(\sum_{m=0}^{\infty} \frac{s^m A^m}{m!} \right) \tag{181}$$

Multiplicando termo a termo, temos:

$$= \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} \frac{A^n A' A^m}{n! m!} \int_0^1 ds \, (1-s)^n s^m \tag{182}$$

A integral pode ser resolvida usando a identidade:

$$\int_0^1 (1-s)^n s^m \, ds = \frac{n! m!}{(n+m+1)!} \tag{183}$$

Substituindo:

$$\int_0^1 ds \; e^{(1-s)A} \frac{dA}{dt} e^{sA} = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} \frac{1}{(n+m+1)!} A^n A' A^m \tag{184}$$

Comparando as equações (179) e (184), obtemos:

$$\frac{d}{dt} e^{A(t)} = \int_0^1 ds \; e^{(1-s)A} \frac{dA}{dt} e^{sA} \tag{185}$$

Multiplicando ambos os lados da equação (185) à esquerda por e^{-A}, temos:

$$\begin{aligned} e^{-A(t)} \frac{d}{dt} e^{A(t)} &= \int_0^1 ds \; e^{-sA} A' e^{sA} \\ &= \int_0^1 ds \; \left(A' + \sum_{n=1}^{\infty} \frac{(-1)^n s^n}{n!} (\mathsf{ad}_A)^n (A') \right) \\ &= \left(A' s + \sum_{n=1}^{\infty} \frac{(-1)^n s^{n+1}}{(n+1)!} (\mathsf{ad}_A)^n (A') \right) \Bigg|_0^1 \\ &= A' + \sum_{n=1}^{\infty} \frac{(-1)^n}{(n+1)!} (\mathsf{ad}_A)^n (A') \end{aligned} \tag{186}$$

Essa identidade será útil para o desenvolvimento da fórmula de Baker-Campbell-Hausdorff.

Queremos agora encontrar $Z(t)$ tal que:

$$e^{tA} e^{tB} = e^{Z(t)} \tag{187}$$

Derivando ambos os lados:

$$e^{-tB} e^{-tA} \frac{d}{dt} (e^{tA} e^{tB}) = e^{-Z(t)} \frac{d}{dt} e^{Z(t)} \tag{188}$$

Calculando o lado esquerdo:

$$\begin{aligned} e^{-tB}e^{-tA}\frac{d}{dt}(e^{tA}e^{tB}) &= e^{-tB}e^{-tA}(Ae^{tA}e^{tB} + e^{tA}Be^{tB}) \\ &= e^{-tB}Ae^{tB} + B \\ &= A + B + \sum_{n=1}^{\infty}\frac{(-1)^n t^n}{n!}(\mathsf{ad}_B)^n(A) \end{aligned} \tag{189}$$

Já o lado direito da equação (188) pode ser expandido usando (186):

$$e^{-Z(t)}\frac{d}{dt}e^{Z(t)} = Z' + \frac{1}{2}[Z', Z] + \cdots \tag{190}$$

Assumindo uma expansão de Taylor para $Z(t)$:

$$Z(t) = Z_1 t + Z_2 t^2 + Z_3 t^3 + \ldots \tag{191}$$

$$Z' = Z_1 + 2tZ_2 + 3t^2 Z_3 + \ldots \tag{192}$$

Calculando o comutador $[Z', Z]$, temos:

$$\begin{aligned} [Z', Z] &= [Z_1 + 2tZ_2 + 3t^2 Z_3 + \ldots, Z_1 t + Z_2 t^2 + Z_3 t^3 + \ldots] \\ &= t^2[Z_1, Z_2] + 2t^2[Z_2, Z_1] + \mathcal{O}(t^3) \\ &= -t^2[Z_1, Z_2] + \mathcal{O}(t^3) \end{aligned} \tag{193}$$

Substituindo tudo, temos:

$$A + B + t[A, B] + \frac{t^2}{2}[B, [B, A]] + \mathcal{O}(t^3) = Z_1 + 2tZ_2 + 3t^2 Z_3 - \frac{t^2}{2}[Z_1, Z_2] + \mathcal{O}(t^3) \tag{194}$$

Comparando os dois lados ordem a ordem:

$$\begin{cases} Z_1 = A + B \\ Z_2 = \frac{1}{2}[A, B] \\ Z_3 = \frac{1}{12}\left([A, [A, B]] + [B, [B, A]]\right) \end{cases} \tag{195}$$

Portanto, obtemos a identidade de Baker-Campbell-Hausdorff:

$$e^{tA}e^{tB} = \exp\left(tA + tB + \frac{t^2}{2}[A, B] + \frac{t^3}{12}([A, [A, B]] + [B, [B, A]]) + \ldots\right) \tag{196}$$

Particularmente, para $t = 1$:

$$e^A e^B = \exp\left(A + B + \frac{1}{2}[A, B] + \frac{1}{12}([A, [A, B]] + [B, [B, A]]) + \ldots\right) \tag{197}$$