

Universidade Federal de São Carlos – UFSCar

Centro de Ciências Exatas e de Tecnologia  
Departamento de Física

Gabriel Pedro Lima Moysés Fernandes

# **Algoritmo Quântico para o Problema de Gerenciamento de Estoque**

São Carlos

2023



Gabriel Pedro Lima Moysés Fernandes

# **Algoritmo Quântico para o Problema de Gerenciamento de Estoque**

Dissertação apresentada ao Programa de Pós-Graduação em Física para obtenção do título de Mestre em Física

Orientador: Prof. Dr. Celso Jorge Villas-Bôas  
Coorientador: Paulo César Cunha Bezerra

São Carlos

2023

---

Gabriel Pedro Lima Moysés Fernandes

Algoritmo Quântico para o Problema de Gerenciamento de Estoque/ Gabriel  
Pedro Lima Moysés Fernandes. – São Carlos, 2023–

Orientador: Prof. Dr. Celso Jorge Villas-Bôas

Coorientador: Paulo César Cunha Bezerra

Tese Final de Mestrado – Universidade Federal de São Carlos – UFSCar

Centro de Ciências Exatas e de Tecnologia

Departamento de Física, 2023.

1. Algoritmos Quânticos. 2. Computação Quântica. I. Celso Jorge Villas-Bôas.  
II. Universidade Federal de São Carlos. III. Departamento de Física. IV. Algoritmo  
Quântico para o Problema de Gerenciamento de Estoque.

---



*À minha família.*



# Agradecimentos

Desde a graduação, acreditei que o mestrado seria o momento acadêmico que culminaria na minha independência como pesquisador, no qual desenvolveria as habilidades necessárias para que pudesse realizar a pesquisa científica por conta própria. Imaginei que o mestrado seria uma jornada quase sempre solitária. Fico feliz em dizer que não poderia estar mais enganado. Precisei do apoio, da compreensão e da ajuda de todos vocês que cito a seguir. Sem vocês, não poderia ter chegado aqui.

Agradeço ao meu orientador, Prof. Dr. Celso Jorge Villas-Bôas, que desde a minha graduação acredita na minha capacidade de realizar os projetos aos quais me proponho, mesmo em momentos onde eu mesmo não tive tanta certeza assim. Agradeço por sempre lutar para que possamos ter as oportunidades de seguir nossos sonhos, superando o que achamos ser nossos próprios limites.

Agradeço ao meu co-orientador, Paulo Bezerra, pela paciência em me explicar tantas vezes em detalhes o funcionamento do depósito de uma fábrica e entender que existem jargões técnicos que eu inevitavelmente reduzo para os nomes mais fáceis de lembrar. Agradeço também por ter embarcado conosco nesse projeto onde indústria e academia apertam as mãos.

Agradeço a todos os colegas do grupo de pesquisa, Alexandre, Amanda, Matheus e Nicolás, por todas as discussões acadêmicas, pela contribuição com tantas ideias e por comparecerem em todas as reuniões emergenciais que surgiam nos horários mais inesperados para tratar, aqui e ali, de um sinal que desapareceu em uma conta ou de um código que não estava funcionando conforme previsto.

Agradeço a minha família por sempre me apoiar em meus sonhos. Especialmente, agradeço as três mulheres da minha vida: mãe, tia e avó. Fariam tudo por mim, se eu ousasse pedir. Agradeço por comemorarem as minhas presenças e compreenderem as minhas ausências. Reservo essa linha para agradecer especialmente à minha mãe, Janaina, que me ensinou a dar meu melhor em tudo que faço.

Agradeço a Ana Carlyne, minha namorada, pelo apoio, pelo suporte e pelo carinho constantes. Por compartilhar comigo cada segundo dessa breve jornada. Por ter feito nossas as minhas alegrias e preocupações, por ter ouvido meus devaneios sobre hamiltonianos e problemas de otimização e por sempre, sempre, estar ao meu lado. Agradeço também à família da Ana, por me acolher em diversos momentos em que tudo que eu precisava era de uma xícara de café.

Agradeço aos meus amigos tanto pela genuína amizade quanto pelos diversos

momentos em que precisei de vocês durante a realização desse projeto. Ao Alexandre, pelas tantas apresentações que fizemos juntos. Ao Ciro, pelas conversas, os conselhos e as milhares de horas de estudo. Ao Lucas, pela fraternidade. Ao Luís, pelas recomendações musicais e os momentos de descontração. À Amanda e ao Matheus, pelo tanto de paciência que tiveram ao me ajudar com tantos códigos de *Python*. Ao Conan e à Ravena, por me ajudar a revisar todo o texto.

Agradeço à UFSCar e ao Centro de Pesquisas Wernher von Braun como instituições de pesquisa e a todos aqueles que nelas se dedicam todos os dias.

Agradeço ao CNPq pelo fomento dessa pesquisa científica.

Agradeço às políticas públicas e a todos aqueles que lutam por elas, particular ou publicamente.

Agradeço a São Bento, por toda a proteção.

# Resumo

Neste projeto de pesquisa, desenvolvemos um algoritmo quântico em Computação Quântica Adiabática para o Problema de Gerenciamento de Estoque, adaptado para fábricas que utilizam prateleiras gravitacionais como sistema padrão de armazenamento. O objetivo central da tarefa é determinar a alocação ideal para os itens do estoque nas prateleiras do depósito, minimizando os custos operacionais associados à logística de entrada e saída de itens e respeitando as restrições envolvidas. Para isso, baseados no padrão de ocorrência dos itens nas demandas da fábrica, elaboramos uma estratégia em duas fases que busca determinar qual a forma mais eficiente de distribuir os itens entre as prateleiras e construímos um conjunto de Hamiltonianos Problemas que permitem implementar a estratégia no modelo de Computação Quântica Adiabática. Por fim, implementamos o algoritmo para um conjunto de instâncias simples nos computadores quânticos da *D-Wave*, uma empresa canadense que é líder no desenvolvimento de dispositivos apropriados para implementação de Computação Quântica Adiabática e utilizam para isso uma heurística conhecida como *Quantum Annealing*.

**Palavras-chaves:** Algoritmo Quântico, Problema do Gerenciamento de Estoque, Computação Quântica Adiabática, *Quantum Annealing*.



# Abstract

In this research project, we have developed a quantum algorithm in Adiabatic Quantum Computing for the Inventory Management Problem, adapted for factories that employ gravity shelving as a standard storage system. The central objective of the task is to determine the optimal allocation for inventory items on the warehouse shelves, minimizing the operational costs associated with item logistics, both inbound and outbound, and adhering to the relevant constraints. To achieve this, based on the occurrence pattern of items in the factory's demands, we have devised a two-phase strategy that aims to determine the most efficient way to distribute items among the shelves and constructed a set of Hamiltonian Problems that enable the implementation of this strategy in the Adiabatic Quantum Computing model. Finally, we implemented the algorithm for a set of simple instances on D-Wave's quantum computers, a Canadian company that leads in the development of devices suitable for Adiabatic Quantum Computing implementation, employing a heuristic known as Quantum Annealing for this purpose.

**Keywords:** Quantum Algorithm, Inventory Management Problem, Adiabatic Quantum Computing, Quantum Annealing.



# Lista de ilustrações

- Figura 1 – Evolução temporal das funções de interpolação  $A(s)$  e  $B(s)$ . No gráfico, as condições de contorno  $A(0) \gg B(0)$  e  $B(1) \gg A(1)$  são genericamente satisfeitas com funções arbitrárias. Se a interpolação for feita de forma suficientemente lenta, o sistema é (ao menos, aproximadamente) levado ao estado fundamental de  $H_P$ , que codifica a solução ótima do problema. 36
- Figura 2 – Representação das topologias presentes nos *quantum annealers* da *D-Wave*. Em (a), a topologia *Chimera*, presente nos chips *D-Wave 2000Q*. Em (b), a topologia *Pegasus*, presente nos chips Advantage. . . . . 39
- Figura 3 – Representação do Problema do Caixeiro Viajante para 6 e 7 cidades. A Figura a) mostra todas as possíveis conexões para o caso de 6 cidades distribuídas aleatoriamente no espaço e a Figura b) mostra a solução ótima encontrada no *quantum annealer*. A Figura c) mostra todas as possíveis conexões para o caso de 7 cidades distribuídas aleatoriamente no espaço e a Figura d) mostra a solução ótima encontrada no *quantum annealer*. . . . . 42
- Figura 4 – Representação de uma caixa-palete. Em fábricas de médio e grande porte, caixas-paletes são utilizadas como unidade padrão de armazenamento. O conteúdo dessas caixas-paletes são insumos requeridos pelas linhas de produção na montagem dos produtos finais. . . . . 48
- Figura 5 – Representação de um conjunto de prateleiras gravitacionais. Em fábricas de médio e grande porte, prateleiras gravitacionais são adotadas para armazenar os paletes no depósito. Na ilustração, a empilhadeira do lado esquerdo introduz uma caixa-palete em uma prateleira gravitacional (trilho), que deslizará até encostar nos demais paletes ali alocados, enquanto a empilhadeira do lado direito faz a retirada de um outro palete requisitado pela linha de produção. . . . . 48
- Figura 6 – As dimensões (em milímetros) dos três tipos de paletes que podem ser armazenados no depósito da fábrica são apresentadas. A largura é representada como a direção de encaixe da empilhadeira. . . . . 50
- Figura 7 – As dimensões (em milímetros) dos três modelos de prateleiras disponíveis no estoque são ilustradas. Nos três modelos, a vista lateral apresenta uma plataforma com 5 prateleiras gravitacionais, enquanto a vista frontal mostra um conjunto dessas plataformas, onde apenas o primeiro palete é visível e pode ser removido pela empilhadeira. . . . . 51

Figura 8 – Espectro do Hamiltoniano gerado na Fase de Agrupamento para o caso em que $N = 3$ paletes devem ser distribuídos em $M = 2$ prateleiras com $L_m = 2$ espaços cada. Os pontos destacados em vermelho indicam os 10 estados de maior ocorrência em uma amostra. . . . .	69
Figura 9 – Gráfico de energia (barra vermelha) e número de ocorrências (barra azul) dos 10 estados mais medidos na amostra. Os dois estados mais presentes na amostra, de <i>bitstrings</i> (em decimal) 408 e 612, correspondem ao estado fundamental degenerado de $H_{agrup}$ , observados 25 e 20 vezes, respectivamente, com energia igual a 1.847521. . . . .	70
Figura 10 – Espectro do Hamiltoniano gerado na Fase de Agrupamento para o caso em que $N = 4$ paletes devem ser distribuídos em $M = 2$ prateleiras com $L_m = 2$ espaços cada. Os pontos destacados em vermelho indicam os 10 estados de maior ocorrência em uma amostra. . . . .	71
Figura 11 – Gráfico de energia (barra vermelha) e número de ocorrências (barra azul) dos 10 estados mais medidos na amostra. Os dois estados mais presentes na amostra, de <i>bitstrings</i> (em decimal) 1632 e 2448, correspondem ao estado fundamental degenerado de $H_{agrup}$ , observados 27 e 12 vezes, respectivamente, com energia igual a 2.7. . . . .	71
Figura 12 – No Painel, determinamos o número de <i>shots</i> necessários para alcançar a solução ótima da Fase de Agrupamento em (a) para $N = 4$ , em (b) para $N = 5$ e em (c) para $N = 6$ . A linha pontilhada em vermelho representa a energia do estado fundamental do Hamiltoniano de Agrupamento, obtida a partir de uma busca exaustiva no espectro do hamiltoniano. . . . .	73
Figura 13 – No Painel, determinamos o número de <i>shots</i> necessários para alcançar a solução ótima da Fase de Agrupamento em (a) para $N = 7$ , em (b) para $N = 8$ e em (c) para $N = 9$ . A linha pontilhada em vermelho representa a energia do estado fundamental do Hamiltoniano de Agrupamento, obtida a partir de uma busca exaustiva no espectro do hamiltoniano. . . . .	74
Figura 14 – Análise do tempo e do número de qubits físicos necessários para a implementação da Fase de Agrupamento no <i>quantum annealer Advantage 4.1</i> da <i>D-Wave</i> para o caso estudado. Em (a) o tempo necessário utilizando o número de shots determinado para alcançar a solução ótima pelo menos uma vez e em (b) o número de qubits físicos utilizados em comparação com o número de qubits lógicos para cada instância. . . . .	76

# Sumário

1	INTRODUÇÃO . . . . .	17
2	OBJETIVOS . . . . .	23
3	INTRODUÇÃO AOS PROBLEMAS DE OTIMIZAÇÃO . . . . .	25
3.1	Problema do Caixeiro Viajante . . . . .	26
3.2	Problema da Mochila . . . . .	27
3.3	Problema do Roteamento de Veículos com Capacidade . . . . .	29
3.4	Limitações das Heurísticas Clássicas e a Computação Quântica . . . . .	30
4	INTRODUÇÃO À COMPUTAÇÃO QUÂNTICA ADIABÁTICA . . . . .	33
4.1	Classical e Quantum Annealing . . . . .	33
4.2	<i>D-Wave</i> e <i>Quantum Annealers</i> . . . . .	36
4.3	<i>Quantum Annealing</i> para o Problema do Caixeiro Viajante . . . . .	39
4.4	<i>Quantum Annealing</i> para o Problema da Mochila . . . . .	41
4.5	Breve Discussão sobre o <i>Quantum Annealing</i> para o Problema do Roteamento de Veículos . . . . .	44
4.6	<i>Quantum Annealing</i> para o Problema de Gerenciamento de Estoque . . . . .	45
5	PROBLEMA DE GERENCIAMENTO DE ESTOQUE . . . . .	47
5.1	Um Exemplo Real . . . . .	49
5.2	Problema das Múltiplas Mochilas Quadráticas . . . . .	53
5.2.1	Problema das Múltiplas Mochilas Quadráticas Necessárias . . . . .	55
6	ALGORITMO QUÂNTICO PARA O PROBLEMA DE GERENCIAMENTO DE ESTOQUE . . . . .	57
6.1	Adaptando o RMQKP para o Gerenciamento de Estoque . . . . .	57
6.2	Obtenção dos Parâmetros via <i>Machine Learning</i> . . . . .	60
6.3	Estratégia em Duas Fases para o Problema de Gerenciamento de Estoque . . . . .	61
6.4	Algoritmo Quântico para o Problema de Gerenciamento de Estoque . . . . .	62
6.5	Implementação do Algoritmo Quântico para o Problema de Gerenciamento de Estoque em <i>Quantum Annealing</i> . . . . .	66
6.5.1	Depósito com Duas Prateleiras Gravitacionais, com $L_m = 2$ . . . . .	67
6.5.2	Depósito com Três Prateleiras Gravitacionais, com $L_m = 3$ . . . . .	72
7	CONCLUSÕES E PRÓXIMOS PASSOS . . . . .	77

**REFERÊNCIAS** ..... 81

# 1 Introdução

Problemas de otimização combinatória são frequentemente encontrados na economia e na indústria, envolvendo processos ligados a tomadas de decisão e planejamento estratégico. A natureza aplicada desses problemas traz consigo grande interesse comercial para o desenvolvimento de algoritmos capazes de alcançar soluções ótimas ou aproximadas, atraindo investimentos significativos nas mais diversas áreas. Exemplos dessas áreas incluem o mercado financeiro, onde tarefas como a otimização de portfólios de ativos desempenham um papel crucial [1], e o setor de comércio, onde tarefas associadas à logísticas de entrega, corte e empacotamento são de extrema relevância [2].

Em termos gerais, um problema de otimização específico de uma empresa, campo de estudo ou setor comercial frequentemente pode ser decomposto em partes que se relacionam com outros problemas já amplamente estudados e reconhecidos na literatura. Por exemplo, o Problema do Caixeiro Viajante [3] e o Problema da Mochila [4], que estão intimamente ligados ao tema dessa dissertação, podem ser identificados por trás de uma grande variedade de aplicações, como no roteamento de veículos [5] e na otimização de portfólios em mercados financeiros [6].

Por sua ampla aplicabilidade, a literatura de ambos os problemas supracitados é extensa e diversas estratégias podem ser empregadas para resolvê-los até certo limite. No entanto, não são conhecidos algoritmos que sejam simultaneamente precisos e eficientes ao abordar instâncias maiores e mais complexas dessas tarefas, sendo este justamente o cenário de interesse comercial. Essa lacuna decorre da limitação da computação clássica em lidar com o crescimento explosivo do espaço de soluções dos problemas de otimização combinatória à medida que o tamanho das instâncias cresce [7]. Consequentemente, quantias significativas são investidas na busca por alternativas capazes de fornecer soluções satisfatórias mesmo diante da complexidade intrínseca a esses problemas.

Mais recentemente, considerando os avanços que a Computação Quântica têm trazido em outros campos da matemática ao utilizar fenômenos como a superposição e o emaranhamento para obter vantagens no processamento de dados, algoritmos quânticos desenvolvidos para abordar problemas de otimização combinatória têm recebido grande destaque dentro da área acadêmica [8]. Há, nesse sentido, grande expectativa tanto por parte da comunidade científica quanto da indústria de que, em um futuro próximo, sejam desenvolvidos algoritmos que utilizem fenômenos quânticos para superar os atuais algoritmos clássicos em eficiência e qualidade das soluções obtidas para esse tipo de tarefa, impactando fortemente muitos segmentos da sociedade.

A história da Computação Quântica começa na década de 1980, quando o físico

norte-americano R. Feynman propõe o uso de sistemas quânticos para simular outros sistemas físicos [9]. Desse ponto de partida, foram desenvolvidos uma série de algoritmos que utilizavam fenômenos quânticos na realização de determinadas tarefas, alcançando ganhos significativos na eficiência de processamento, isto é, *speedups*.

Um exemplo notável dessa conquista é o algoritmo de Shor [10] proposto por P. Shor em 1994, capaz de fatorar números inteiros extremamente grandes em seus fatores primos usando um número de passos que cresce de forma polinomial com o número de dígitos do número a ser fatorado, em contraste com o crescimento exponencial dos algoritmos clássicos conhecidos. Outro exemplo é o algoritmo de Grover [11], proposto por L. K. Grover em 1996, que efetua buscas em bancos de dados de maneira consideravelmente mais eficiente do que as estratégias clássicas conhecidas. Nesse sentido, já nos anos de 1990, tornou-se evidente que a Computação Quântica tinha o potencial de oferecer vantagens na resolução de determinadas tarefas, dando então início a um intenso investimento na busca por algoritmos quânticos que superassem seus equivalentes clássicos em eficiência.

Esse investimento levou a grandes evoluções na área, resultando no desenvolvimento de novos e melhores *hardwares* [12], em debates calorosos acerca da chamada vantagem quântica [13, 14] e na criação de diferentes modelos de Computação Quântica [15], que buscam alcançar *speedups* na realização de tarefas utilizando fenômenos quânticos de maneiras distintas. Entre esses, nos últimos anos, um modelo que tem se destacado na literatura para o desenvolvimento de algoritmos voltados para problemas de otimização é o modelo de Computação Quântica Adiabática [16].

Nos algoritmos de Computação Quântica Adiabática (CQA), as possíveis soluções de um problema de otimização são codificadas no espectro de um hamiltoniano, chamado Hamiltoniano Problema, de tal forma que quão maior a qualidade da solução codificada, menor é a autoenergia associada ao autoestado que a codifica. No estado fundamental, dessa forma, é codificada a a solução ótima da tarefa.

Para implementação desses algoritmos [17], um sistema é preparado no estado fundamental de um Hamiltoniano Inicial, sob o qual é realizada uma evolução até o Hamiltoniano Problema, que codifica o espaço de soluções da tarefa se quer abordar. De acordo com o Teorema Adiabático, se a evolução é feita adiabaticamente, o estado que descreve o sistema ao final do processo é o estado fundamental do Hamiltoniano Problema, alcançando assim a solução ótima da tarefa [16].

Na realização prática dessa implementação, entretanto, existem diversas possíveis fontes de erro [18, 19], incluindo interações com o ambiente e erros de calibração dos componentes que realizam a interpolação entre o Hamiltoniano Inicial e o Hamiltoniano Problema. Como resultado, nem sempre o estado final medido após o processo de interpolação é, de fato, o estado fundamental, podendo haver desvios para mínimos locais, o que traz uma série de considerações sobre a CQA que estão inevitavelmente relacionadas ao

---

atual estado da arte dos *hardwares* disponíveis, como ocorre (cada qual a seu modo) nos outros modelos de Computação Quântica.

Atualmente, algoritmos de CQA podem ser implementados nos computadores quânticos da empresa canadense *D-Wave Systems* [20]. Pioneira e líder no desenvolvimento de dispositivos apropriados para implementação de CQA, a empresa utiliza a metaheurística conhecida como *Quantum Annealing* [21] para realizar a interpolação física entre os hamiltonianos que descrevem a configuração inicial e final do sistema, sendo, portanto, tais dispositivos conhecidos como *quantum annealers*.

Na recente literatura que busca explorar o potencial dos *quantum annealers* da *D-Wave*, se destacam algoritmos quânticos e suas implementações para o Problema do Caixeiro Viajante [22], para o Problema da Mochila [23], entre outros. Nesses trabalhos, embora não tenha sido demonstrado que existam vantagens quânticas obtidas pela CQA dado o atual estado da arte dos *quantum annealers*, obteve-se sucesso na implementação de instâncias simples dessas tarefas, sendo que a qualidade das soluções obtidas nesses casos são comparáveis a (e até superam) algumas das heurísticas clássicas mais consolidadas, como apontado na implementação do Problema do Roteamento de Veículos [24].

Nesse projeto de mestrado, que surge como resultado de uma parceria entre o *Grupo de Óptica Quântica e Informação Quântica (Departamento de Física - Universidade Federal de São Carlos)* e o *Centro de Pesquisas Avançadas Wernher von Braun*, temos como objetivo principal desenvolver um algoritmo em Computação Quântica Adiabática para abordar o Problema de Gerenciamento de Estoque. O propósito central dessa tarefa é determinar a alocação ideal dos paletes do estoque de uma fábrica nos sistemas de armazenamento do depósito, minimizando os custos operacionais associados à logística de entrada e saída de itens [25] e respeitando as restrições envolvidas, que dependem das necessidades de cada fábrica.

De forma a lidar com o cenário específico que nos interessa, em uma parceria mediada pelo *Centro de Pesquisas Avançadas Wernher von Braun*, adaptamos o problema para atender as necessidades de uma fábrica real que opera em território nacional. No depósito dessa fábrica, cujo nome preservaremos em sigilo por questões de confidencialidade dos dados que serão apresentados no decorrer do texto, prateleiras gravitacionais são utilizadas como sistema padrão de armazenamento. Essas prateleiras, de forma simplificada, funcionam como trilhos inclinados que utilizam da força gravitacional para automatizar o movimento dos paletes quando uma operação de retirada é realizada [26].

Além disso, essas prateleiras operam de acordo com o princípio de gerenciamento FIFO<sup>1</sup> [27], garantindo que a ordem de retirada dos paletes alocados em uma mesma prateleira siga a ordem cronológica de inserção, evitando perdas por deterioração. Essa é

---

<sup>1</sup> *First In, First Out*. Em tradução livre, *Primeiro a entrar, primeiro a sair*.

uma restrição que deve ser levada em conta no desenvolvimento de uma estratégia que busca determinar a alocação ideal dos paletes nas prateleiras gravitacionais da fábrica, de tal forma que os custos envolvidos nas futuras operações do depósito sejam reduzidos. Vale ressaltar que, em uma extensa busca pela literatura, não foram encontradas estratégias (clássicas ou quânticas) que realizassem a tarefa aqui proposta.

Portanto, o objetivo fundamental dessa dissertação é desenvolver um algoritmo quântico para o Problema de Gerenciamento de Estoque, adaptado para fábricas que adotam prateleiras gravitacionais como sistema padrão de armazenamento, em Computação Quântica Adiabática. Para isso, nossa abordagem envolve a formulação de uma estratégia em duas fases para otimizar a alocação dos paletes nas prateleiras e, posteriormente, para cada uma dessas etapas, a elaboração de um Hamiltoniano Problema. Determinadas as fases da nossa estratégia e os Hamiltonianos Problemas que as codificam, será possível implementar o algoritmo quântico nos *quantum annealers* da *D-Wave*.

Para uma melhor compreensão da estratégia desenvolvida e dos resultados obtidos neste projeto, organizamos o texto que segue da seguinte forma:

- No Capítulo 2, apresentamos os objetivos do projeto.
- No Capítulo 3, apresentamos uma sucinta introdução sobre problemas de otimização combinatória, na qual nos aprofundamos em tarefas amplamente estudadas na literatura e que serão especialmente úteis nos capítulos finais dessa dissertação, em específico, o Problema do Caixeiro Viajante e o Problema da Mochila. Mostramos também como essas tarefas podem ser combinadas para compor problemas de otimização em contextos de aplicabilidade mais evidente, como o Problema do Roteamento de Veículos, atraindo assim o interesse comercial e industrial. Finalizamos a seção discutindo como o crescimento explosivo do espaço de soluções dos problemas de otimização combinatória é até o presente momento um entrave para o desenvolvimento de algoritmos precisos e eficientes para estas tarefas, justificando a busca por novas abordagens, como a Computação Quântica Adiabática.
- No Capítulo 4, apresentamos uma introdução mais detalhada sobre a Computação Quântica Adiabática, nos aprofundando em suas bases teóricas, que são fundamentadas no Teorema Adiabático. Em sequência, discutimos aspectos relevantes acerca dos *quantum annealers* da *D-Wave*, como a metaheurística utilizada para realizar a interpolação entre o Hamiltoniano Inicial e o Hamiltoniano Problema (isto é, o *Quantum Annealing*), as características e limitações dos dispositivos mais avançados disponíveis, e o tipo específico de Hamiltoniano Problema que se pode implementar nesses computadores, que são formulados em uma forma restrita de operador conhecida como QUBO.

Por fim, com o intuito de criar bases para o desenvolvimento dos Hamiltonianos Problemas para o Problema de Gerenciamento de Estoque, construímos estes operadores para o Problema do Caixeiro Viajante e para o Problema da Mochila, baseando-nos na literatura disponível [28]. Como prova de conceito, implementamos o Problema do Caixeiro Viajante para instâncias com 6 e 7 cidades, obtendo as rotas mais curtas, como confirmado pela implementação do Método da Força Bruta.

Outras discussões pertinentes, como uma estratégia para a implementação do Problema do Roteamento de Veículos nesses dispositivos, também compõe a seção.

- No Capítulo 5, apresentamos em detalhes o Problema do Gerenciamento de Estoque, oferecendo um panorama geral do funcionamento do depósito da fábrica parceira desse projeto. Para isso, apresentamos as dimensões dos componentes do depósito, como as prateleiras gravitacionais e os modelos de paletes comportados nelas, incluímos uma descrição sobre o princípio FIFO e discutimos as restrições que devem ser respeitadas na elaboração de uma estratégia para otimização da tarefa de distribuir os paletes nas prateleiras tal que os custos associados sejam minimizados nas futuras operações do depósito. Ainda, fornecemos uma revisão da literatura da área, apontando referências que são consistentes com a nossa conclusão de que não existem trabalhos similares ao nosso para o modelo de fábrica analisada.

Além disso, preparando terreno para a estratégia em duas fases descrita no capítulo seguinte, apresentamos o Problema das Múltiplas Mochilas Quadráticas Necessárias, uma variante do Problema da Mochila que adaptamos para mapear uma das fases da nossa estratégia. A outra fase, mapeada em uma versão adaptada do Problema do Caixeiro Viajante, justifica também a atenção redobrada dada a esses dois problemas desde o início do texto.

- No Capítulo 6, onde se concentram a maior parte dos resultados deste projeto, elaboramos a estratégia em duas fases para o Problema de Gerenciamento de Estoque conforme delineado até aqui.

Na primeira fase da estratégia, que denominamos como Fase de Agrupamento, utilizamos um conjunto de parâmetros gerados via a aplicação de técnicas de *machine learning* sob registros das demandas da fábrica para determinar quais paletes devem ser alocados nas mesmas prateleiras.

Na segunda fase da estratégia, que denominamos como Fase de Inserção, para cada grupo de paletes formado na fase anterior, determinamos a ordem de inserção na prateleira para o qual foram destinados.

Em sequência, para cada uma das fases, construímos um Hamiltoniano Problema que possibilita sua implementação nos *quantum annealers* da *D-Wave*, concluindo assim o objetivo de desenvolver um algoritmo quântico para o Problema de Gerenciamento

de Estoque. Finalmente, de forma a validar seu funcionamento, implementamos o algoritmo nos *quantum annealers* para um conjunto de instâncias simples, onde o depósito está inicialmente vazio e a tarefa se reduz a distribuir um dado número de paletes em um dado número de prateleiras de forma a minimizar os custos associados.

- Por fim, no Capítulo 7, apresentamos o que consideramos os próximos passos da pesquisa e nossas considerações finais.

## 2 Objetivos

O objetivo principal desse projeto é desenvolver um algoritmo quântico para o Problema de Gerenciamento de Estoque, adaptado para fábricas que adotam prateleiras gravitacionais como sistema padrão de armazenamento, em Computação Quântica Adiabática. Para isso, orientamos o projeto tal que nossos objetivos específicos fossem

1. Compreender os formalismos da Computação Quântica Adiabática, bem como o funcionamento e as limitações dos *hardwares* atuais.
2. Levando em conta tais limitações, executar algoritmos quânticos de interesse comercial/industrial nos *quantum annealers* da empresa canadense *D-Wave*, como, por exemplo, o Problema do Caixeiro Viajante.
3. Desenvolver uma estratégia para determinar a alocação ideal dos paletes do estoque de uma fábrica como anteriormente descrita nas prateleiras do depósito.
4. Formular um conjunto de Hamiltonianos Problemas que permitam implementar essa estratégia nos *quantum annealers* da *D-Wave*.
5. Implementar esta estratégia para um caso simples nos *quantum annealers* da *D-Wave*, atestando sua validade.

Desde o início do projeto, portanto, orientamos nossos objetivos específicos de tal forma que, ao cumprí-los, cumpriríamos também com o nosso objetivo central, isto é, de desenvolver um algoritmo quântico como especificado para o Problema de Gerenciamento de Estoque em CQA.



## 3 Introdução aos Problemas de Otimização

Problemas de otimização são uma classe de problemas matemáticos em que se busca encontrar o melhor valor<sup>1</sup> de uma dada função sujeita a um conjunto de restrições [29]. Esses problemas podem ser classificados de várias maneiras, dependendo das características das soluções e das restrições envolvidas. Uma subclasse importante de problemas de otimização é a de otimização combinatória [30].

Nesses problemas, o objetivo é determinar a melhor entre um conjunto finito e discreto de possíveis soluções, que são formadas pelas diferentes combinações das variáveis da tarefa. Nesse sentido, encontrar a melhor solução (ou solução ótima) é encontrar a combinação das variáveis que, respeitando as restrições envolvidas, extremiza a função que caracteriza o problema, dita função objetivo.

Em geral, problemas de otimização combinatória são desafiadores de se resolver devido ao grande número de possíveis soluções e às complexas restrições envolvidas [31]. Graças a sua grande aplicabilidade, no entanto, esses problemas são frequentemente encontrados em diversas áreas, envolvendo processos ligados a tomadas de decisão e planejamento estratégico, por estarem diretamente conectados com a maximização do lucro e eficiência das tarefas a eles relacionados.

Um problema de otimização combinatória específico muitas vezes pode ser decomposto em partes que recaem em problemas mais gerais e possivelmente conhecidos. Por exemplo, o Problema do Caixeiro Viajante [3] e o Problema da Mochila [4], podem ser encontrados por trás de uma grande variedade de logísticas sob os mais diferentes contextos, como no roteamento de veículos em estratégias de entrega e distribuição [5], na otimização de portfólios em mercados financeiros [6] e no gerenciamento de sistemas em cidades inteligentes [32].

A literatura acerca do Problema do Caixeiro Viajante e do Problema da Mochila é vasta e há diversas estratégias que podem ser empregadas para resolvê-los até certo limite [33, 34]. Entretanto, o crescimento explosivo do espaço de soluções desses problemas é um obstáculo para o desenvolvimento de algoritmos que encontrem soluções exatas de forma eficiente. Métodos baseados na busca exaustiva por todas as soluções possíveis, por exemplo, se tornam ineficientes nesses casos [33, 34].

Uma alternativa para lidar com esse desafio é empregar técnicas heurísticas [35], que são estratégias mais inteligentes e não determinísticas para encontrar soluções aproximadas explorando regiões promissoras do espaço de soluções. No entanto, até mesmo as mais bem consolidadas heurísticas têm seu poder de resolução limitado e falham em

---

<sup>1</sup> O melhor valor é definido por um critério de otimização.

ser simultaneamente precisas e eficientes na abordagem de instâncias maiores e mais complexas, o que resulta na perda da qualidade<sup>2</sup> das soluções encontradas ou em tempos inviáveis de processamento, traduzindo-se na ineficiência das tarefas e em prejuízos de grande valor financeiro.

Nesta seção, temos como objetivo apresentar o Problema do Caixeiro Viajante e o Problema da Mochila, pontuando algumas das suas versões mais conhecidas e detalhando o crescimento explosivo do espaço de soluções das tarefas, que limita significativamente o poder de resolução até mesmo das estratégias clássicas mais avançadas disponíveis. Além disso, mostraremos como esses problemas podem ser combinados para compor outros problemas em cenários de aplicabilidade evidente, como o Problema do Roteamento de Veículos [36], justificando assim o interesse comercial e industrial em buscar novas abordagens para superar as atuais limitações computacionais.

### 3.1 Problema do Caixeiro Viajante

O Problema do Caixeiro Viajante (ou *Traveling Salesman Problem*, TSP) é um problema de otimização combinatória que pode ser enunciado como segue [3]: “*Um caixeiro viajante precisa realizar entregas em  $N$  cidades. Qual é a rota que minimiza o custo de uma viagem, passando por todas as cidades exatamente uma vez antes de retornar ao ponto de partida?*”.

O problema acima exposto é classificado como NP-completo [3], sendo que o espaço de soluções da tarefa cresce de forma fatorial com o número de cidades a serem visitadas. Isso significa que este é um problema difícil de ser resolvido do ponto de vista computacional, especialmente quando há um grande número de cidades envolvidas. Os algoritmos exatos existentes empregam estratégias para reduzir essa complexidade, alcançando a solução ótima do Problema do Caixeiro Viajante com uma ordem de complexidade temporal igual a  $O(N^2 2^N)$  [37], o que ainda é um crescimento exponencial em  $N$  e se torna impraticável para instâncias onde um grande número de cidades é considerado.

Matematicamente, o Problema do Caixeiro Viajante pode ser posto com maior generalidade ao considerar um grafo  $G = G(V, E)$ , onde  $V = \{1, \dots, N\}$  é o conjunto de vértices que representam as cidades e  $E$  é o conjunto de arestas que representam todas as possíveis conexões entre os vértices. À aresta que conecta o  $i$ -ésimo ao  $j$ -ésimo vértice, associamos o peso  $W_{ij}$ . Dessa forma, o problema de minimização do custo da rota pode ser escrito como: “*A partir de um vértice de escolha, qual o caminho que liga todos os vértices e retorna ao ponto de partida (formando um ciclo) tal que a soma  $\sum W_{ij}$  seja mínima?*”.

O peso  $W_{ij}$  pode ser definido de diferentes formas. É possível, em uma interpretação

<sup>2</sup> Define-se como qualidade de uma solução a distância entre esta e a solução ótima. Quanto menor a distância, maior a qualidade.

mais imediata, defini-lo como a distância entre as cidades  $i$  e  $j$ , que pode ser diretamente obtida a partir da posição geográfica das cidades. Como consequência dessa interpretação, uma matriz com todos os pesos  $W_{ij}$  é simétrica e dizemos que o grafo é não-direcionado [38]. O número de conexões (arestas) entre as cidades é  $N(N-1)/2$  e existem  $(N-1)!/2$  possíveis rotas.

Outra alternativa seria definir o peso  $W_{ij}$  como o valor da viagem entre as cidades  $i$  e  $j$  - incluindo pedágios, desvios e demais custos, o que possivelmente implicaria em  $W_{ij} \neq W_{ji}$  para dado  $(i, j) \in E$  fixos, e dizemos que o grafo é direcionado [38]. Nesse caso, o número de conexões entre as cidades é igual a  $N(N-1)$  e existem  $(N-1)!$  possíveis rotas.

Muito além de sua elaboração simples, o Problema do Caixeiro Viajante pode ser encontrado por trás de tarefas complexas e de ampla importância comercial, industrial e também social. Dentre estas tarefas estão [5, 39, 40]: o planejamento de rotas em logísticas de coleta, entrega e distribuição, o planejamento de rotas de tráfego, o projeto de redes de distribuição elétrica, a otimização de redes de telecomunicação e a esquematização de circuitos eletrônicos.

Ainda, embora o Problema do Caixeiro Viajante esteja presente em diversas logísticas e contextos, o crescimento fatorial de seu espaço de soluções é um entrave para o desenvolvimento de algoritmos capazes de abordá-lo de forma eficiente para instâncias maiores e mais complexas. Dessa forma, as estratégias e heurísticas utilizadas para abordar a tarefa podem ser classificadas em duas categorias [41]: aquelas que são exatas, mas ineficientes, como a Programação Dinâmica [37], e aquelas que são eficientes, mas não exatas, como o Algoritmo Genético [42].

## 3.2 Problema da Mochila

O Problema da Mochila (ou *Knapsack Problem*, KP) é um problema de otimização combinatória que pode ser enunciado como segue [43]: “Dada uma lista de  $N$  itens únicos, cada item possuindo um determinado valor e peso, quais objetos devem ser colocados em uma mochila tal que o peso total seja menor ou igual a um dado limite  $L$  e o valor carregado seja maximizado?”.

Assim como o Problema do Caixeiro Viajante, o Problema da Mochila é classificado como NP-completo [4], sendo que seu espaço de soluções cresce exponencialmente com o número de itens, existindo  $2^N$  possíveis configurações para a mochila. Os algoritmos exatos existentes permitem reduzir esta complexidade, alcançando a solução ótima com uma ordem de complexidade temporal igual a  $O(N * L)$  [44], o que ainda leva a tempos de execução significativos para instâncias maiores e mais complexas, nos quais os parâmetros de entrada são muito grandes.

Matematicamente, o Problema da Mochila pode ser posto com maior generalidade da seguinte forma: “*Considere uma lista de  $N$  itens únicos, enumerados pelo índice  $\alpha$ . Cada item possui um determinado valor  $v_\alpha$  e peso  $w_\alpha$ . Considere uma mochila cujo limite máximo de peso é  $L$ . Se  $x_\alpha$  é uma variável binária que representa se o item está (1) ou não (0) contido na mochila, então o peso total carregado é dado por  $W = \sum_{\alpha=1}^N w_\alpha x_\alpha$  e o valor total da mochila é dado por  $V = \sum_{\alpha=1}^N v_\alpha x_\alpha$ . A tarefa consiste em maximizar o valor total da mochila  $V$ , respeitando à restrição  $W \leq L$ ”.*

A definição do Problema da Mochila acima exposta é conhecida como Problema da Mochila Binária (ou *0-1 Knapsack Problem*, 0-1 KP), que restringe o número de cópias de cada item na mochila a zero ou uma unidade. Versões alternativas [34] incluem o Problema da Mochila com Repetições Limitadas (*Bounded Knapsack Problem*, BKP), que remove a restrição de que apenas um item de cada tipo pode ser posto na mochila, mas ainda restringe o número de cópias do mesmo item a um dado limite  $\ell$  tal que  $x_\alpha \in \{0, 1, \dots, \ell\}$ , e o Problema da Mochila com Repetições Ilimitadas (*Unbounded Knapsack Problem*, UKP), que anula a existência de um limite superior para o número de possíveis cópias.

Além das modificações no número de cópias idênticas, há ainda versões mais gerais e complexas da tarefa que são amplamente exploradas na modelagem matemática de muitos problemas práticos. Exemplos dessas versões são [45]:

- Problema das Múltiplas Mochilas (*Multiple Knapsack Problem*, MKP): Onde existem várias mochilas e os itens devem ser distribuídos de forma a maximizar a soma do valor total de todas as mochilas.
- Problema da Mochila Quadrática (*Quadratic Knapsack Problem*, QKP): Onde a contribuição de cada item no valor total de uma mochila não é linear e depende das combinações com outros itens.
- Problema da Mochila Multidimensional (*Multidimensional Knapsack Problem*, MdKP): Onde os itens possuem múltiplas dimensões (por exemplo, peso, volume e valor) e a mochila possui restrições em cada dimensão.

De forma geral, a versatilidade das variações do Problema da Mochila justifica sua importância na modelagem de uma ampla gama de aplicações de cunho socio-governamental, comercial, econômico e industrial. Dentre estas aplicações estão [4, 34, 45, 46]: a distribuição de recursos em sistemas inteligentes, a seleção de ativos em portfólios de investimento e a otimização de logísticas de entrega, produção, corte e embalagem.

No entanto, assim como ocorre com o Problema do Caixeiro Viajante, ainda que as variações do Problema da Mochila sejam de extrema relevância em diferentes logísticas e contextos, o crescimento explosivo do espaço de soluções (presente em todas as versões)

representa um desafio latente para o desenvolvimento de algoritmos que lidem de forma eficiente com instâncias maiores e mais complexas. Nesse sentido, a máxima continua válida: os algoritmos exatos, como o *Branch and Bound* [47], revelam-se ineficientes, enquanto os algoritmos eficientes, como o Algoritmo Genético [48] e a Busca Tabu [49], não garantem a obtenção da solução ótima.

### 3.3 Problema do Roteamento de Veículos com Capacidade

O Problema do Roteamento de Veículos com Capacidade (*Capacitated Vehicle Routing Problem*, CVRP) é um problema de otimização combinatória que pode ser enunciado como segue [50]: “*Uma empresa possui uma frota de veículos com capacidade limitada e um conjunto de clientes distribuídos geograficamente. Partindo de um único depósito, qual é o conjunto de rotas que permite que a frota de veículos atenda às demandas de todos os clientes e retorne à origem, minimizando os custos envolvidos?*”.

O CVRP é uma variação do Problema de Roteamento de Veículos (*Vehicle Routing Problem*, VRP), um problema classificado como NP-difícil [50] e proposto pela primeira vez em 1959 [51]. O objetivo principal do VRP é encontrar um conjunto de rotas que permita à empresa transportar as demandas de todos os clientes, utilizando uma frota de veículos, de forma a minimizar os custos totais. Isso implica em decidir quais clientes (e, portanto, quais demandas) cada veículo deve atender, minimizando o custo total das viagens.

A principal diferença entre o VRP e o CVRP é que o VRP não considera a restrição de capacidade dos veículos, ou seja, assume-se que todos os veículos possuem capacidade infinita (ou pelo menos suficiente) para transportar às demandas dos clientes [52]. Por outro lado, ao considerar a restrição de capacidade, o CVRP se torna uma abordagem mais apropriada para empresas cujas mercadorias ocupam espaços consideráveis nos veículos ou quando o número de entregas em uma localização específica excede a capacidade de um único veículo devido às dimensões dos produtos.

Matematicamente, o Problema do Roteamento de Veículos com Capacidade pode ser posto com maior generalidade ao considerar um grafo  $G = G(V, E)$ , onde  $V = \{1, \dots, N\}$  é o conjunto de vértices que representam as posições dos clientes distribuídos geograficamente e  $E \in (i, j)$  é o conjunto de arestas que representam as conexões entre os clientes. À aresta que conecta o  $i$ -ésimo ao  $j$ -ésimo vértice, associamos um custo  $D_{ij}$ . Além disso, assume-se que existem  $M$  veículos com capacidade  $Q$  estacionados no depósito<sup>3</sup> e que cada cliente tem uma demanda  $q$  que precisa ser atendida. Assim, o CVRP é definido como a tarefa de encontrar o conjunto de rotas em que:

<sup>3</sup> Adicionamos um vértice representando o depósito em  $V$ , juntamente com as arestas que conectam os clientes ao depósito em  $E$ .

- (i) Cada cliente em  $V$  é visitado apenas uma vez por um único veículo,
- (ii) Todos os veículos começam e terminam no depósito,
- (iii) A soma das demandas dos clientes em uma rota não excede a capacidade  $Q$ , e
- (iv) A soma dos custos de todas as rotas é minimizada, otimizando a tarefa.

Em suma, o Problema do Roteamento de Veículos com Capacidade tem como objetivo determinar o conjunto de melhores rotas para que uma frota de veículos possa atender às demandas dos clientes, levando em consideração a restrição de capacidade e minimizando a soma dos custos das viagens. No entanto, devido ao grande número de variáveis e restrições envolvidas e ao crescimento superexponencial do espaço de soluções do problema, uma abordagem direta para resolver a tarefa é computacionalmente inviável.

Para contornar esta inviabilidade, uma estratégia amplamente utilizada na literatura [5, 50, 53] é dividir a tarefa em duas fases: uma relacionada à distribuição das demandas entre os veículos da frota (Fase de Clusterização) e outra relacionada à otimização das rotas individuais de cada veículo (Fase de Roteamento).

Na implementação da estratégia, é comum mapear a Fase de Clusterização em um Problema de Múltiplas Mochilas, onde as mochilas representam os veículos com capacidade limitada e os itens representam as demandas a serem entregues. Por sua vez, a Fase de Roteamento pode ser mapeada em um Problema do Caixeiro Viajante, onde o conjunto de cidades é formado pelos clientes e pelo depósito, e as distâncias são determinadas pelos custos de viagem entre estes.

Dessa forma, o Problema do Roteamento de Veículos é um exemplo de como o Problema do Caixeiro Viajante e o Problema da Mochila podem ser combinados para descrever problemas complexos com aplicabilidade evidente. Nesse sentido, as heurísticas utilizadas para abordar estes problemas podem ser combinadas de maneira a abordar o CVRP em fases. No entanto, vale ressaltar que embora a abordagem em etapas seja mais viável do que uma abordagem direta, os problemas de otimização nos quais as fases são mapeadas também possuem crescimento explosivo do espaço de soluções<sup>4</sup> e, portanto, instâncias maiores e mais complexas continuam sendo difíceis de se abordar precisa e eficientemente.

### 3.4 Limitações das Heurísticas Clássicas e a Computação Quântica

Até aqui, discutimos como problemas de otimização combinatória, como o Problema do Caixeiro Viajante e o Problema da Mochila, podem ser usados para modelar aplicações

<sup>4</sup>  $O(M * 2^N)$  para o Problema das Múltiplas Mochilas e  $O(n!)$  para o Problema do Caixeiro Viajante, onde  $n$  denota o número de clientes em uma rota. As complexidades apontadas são aquelas para o Método da Força Bruta, mas podem ser reduzidas com o uso de estratégias mais inteligentes como apontado anteriormente.

ligadas a processos de decisão e planejamento estratégico dentro da indústria, da economia e de outros setores de grande importância social. Além disso, abordamos como o crescimento explosivo do espaço de solução destes problemas limita o poder de resolução das heurísticas mais bem consolidadas, resultando em soluções de qualidade inferior e prejuízos financeiros significativos.

Atualmente, impulsionadas pelo investimento dos diversos setores que buscam superar as limitações das estratégias clássicas, novas abordagens para essas tarefas estão sendo desenvolvidas, explorando o potencial de tecnologias emergentes, como a Computação Quântica [54]. A Computação Quântica é uma área que utiliza fenômenos de sistemas quânticos, como a superposição e o emaranhamento, para realizar cálculos de forma mais eficiente em comparação com a computação clássica no processamento de certas tarefas [55]. Entre os diferentes modelos de Computação Quântica, destaca-se a Computação Quântica Adiabática [17] como um modelo promissor para o desenvolvimento de algoritmos que abordam problemas complexos de otimização combinatória.

Na Computação Quântica Adiabática [16], as possíveis soluções de um problema de otimização são codificadas no espectro de um hamiltoniano, de modo que quanto maior a qualidade da solução codificada, menor é a autoenergia associada ao autoestado que a codifica. No estado fundamental, portanto, é codificada a solução ótima da tarefa. Em seguida, um sistema é preparado no estado fundamental de um hamiltoniano conhecido, que é então evoluído até o hamiltoniano no qual estão codificadas as soluções do problema de otimização. De acordo com o Teorema Adiabático, se a evolução é feita adiabaticamente, o estado que descreve o sistema ao final do processo é o estado fundamental, no qual está codificada a melhor solução.

Desde 2007, a empresa canadense *D-Wave Systems* [20] desenvolve computadores quânticos apropriados para a implementação de Computação Quântica Adiabática. Por usarem uma metaheurística denominada *Quantum Annealing* para realizar a interpolação entre os hamiltonianos que descrevem a configuração inicial e final do sistema, estes dispositivos são conhecidos como *quantum annealers*. Embora esses dispositivos tenham diversas limitações, eles já foram empregados com sucesso para abordar vários problemas de otimização combinatória, incluindo o Problema do Caixeiro Viajante [22], o Problema da Mochila [23] e o Problema do Roteamento de Veículos [24, 56], previamente citados.

No próximo capítulo, descreveremos como algoritmos formulados no modelo de Computação Quântica Adiabática utilizam fenômenos quânticos para buscar a solução ótima no espaço de soluções de problemas de otimização combinatória. Além disso, vamos pontuar as limitações desses algoritmos, o que inclui uma discussão sobre o atual estado da arte dos *quantum annealers*. Por fim, demonstraremos como o Problema do Caixeiro Viajante e o Problema da Mochila podem ser modelados para serem implementados nestes dispositivos, o que envolve codificar as possíveis soluções das tarefas em um tipo específico

de hamiltoniano conhecido como QUBO, que será elaborado para cada uma dessas tarefas.

# 4 Introdução à Computação Quântica Adiabática

## 4.1 Classical e Quantum Annealing

*Annealing*, que em português pode ser traduzido como recozimento, tem significados similares nos mundos clássico e quântico, denominando heurísticas que se fundamentam em princípios físicos para abordar problemas de otimização combinatória [17].

O *Classical Annealing* é uma heurística inspirada em simulação computacional, que se baseia nos princípios da termodinâmica clássica, para abordar problemas de otimização combinatória [57]. Nessa abordagem, um Hamiltoniano Clássico é formulado para codificar as soluções do problema em termos de energias associadas a estados clássicos, de tal forma que quanto menor é a energia de uma solução, maior é a sua qualidade, sendo a solução de menor energia (associada ao estado fundamental) a solução ótima [58].

Com isso, o processo de *Classical Annealing* se inicia com o sistema preparado em um estado aleatório, frequentemente associado a uma alta energia. Em seguida, a temperatura<sup>1</sup> é reduzida lentamente e de maneira controlada, permitindo que o sistema alcance o equilíbrio termodinâmico em cada etapa [58]. A cada redução da temperatura, um estado na vizinhança do estado atual é aleatoriamente escolhido e sua energia é calculada. Se a variação da energia é negativa (ou nula), então o sistema transita para o novo estado. Caso a variação seja positiva, então uma probabilidade de aceitação que dita se o sistema transita (ou não) para o novo estado é calculada. A probabilidade de aceitação<sup>2</sup> depende do inverso da temperatura e, portanto, à medida que a temperatura é reduzida, a probabilidade do sistema saltar de um estado para o outro diminui. Quando uma transição é realizada, o processo de busca é feito novamente. O número de processos de busca (ou de verificação da aceitação da transição) é predeterminado. Ao fim, do *loop*, a temperatura decresce um passo, seguindo um cronograma de resfriamento [59], e todo o processo é refeito. Ao final de sequenciais iterações, quando um critério de terminação é atingido, o estado final, que corresponde a uma solução de alta qualidade se os parâmetros utilizados estiverem corretamente ajustados, é alcançado [59].

Já o *Quantum Annealing* (QA) é uma heurística fundamentada nos princípios da Computação Quântica Adiabática (CQA) [21]. Por sua vez, para abordar problemas de otimização, a CQA se baseia na interpolação adiabática entre um Hamiltoniano Inicial  $H_I$ ,

<sup>1</sup> A temperatura aqui é um parâmetro numérico controlável, e não uma temperatura física.

<sup>2</sup> A probabilidade de aceitação para um estado de energia maior (e, portanto, com uma solução de qualidade pior) existe como uma forma de escapar de mínimos locais.

cuja configuração de menor energia é conhecida e facilmente preparável, e um Hamiltoniano Problema  $H_P$ , que codifica em seu espectro as soluções do problema de tal forma que quanto menor a autoenergia de um autoestado, maior é a qualidade da solução nele mapeada, tal que o estado fundamental corresponda à solução ótima [16]. Vale notar que  $H_P$  deve ser construído especificamente para cada tarefa que se quer abordar, enquanto o mesmo não ocorre com  $H_I$ .

Nos algoritmos de CQA, o processo físico no qual ocorre a interpolação  $H_I \rightarrow H_P$  pode ser descrito por um hamiltoniano dependente do tempo parametrizado  $s$ ,  $s \in [0, 1]$ , tal que

$$H(s) = A(s)H_I + B(s)H_P, \quad (4.1)$$

onde  $A(s)$  e  $B(s)$  são as funções que descrevem a interpolação, obedecendo às condições de contorno gerais  $A(0) \neq 0$ ,  $B(1) \neq 0$  e  $A(1) = B(0) = 0$  [17]. Se inicialmente o sistema está preparado no estado fundamental (conhecido) de  $H_I$  e a dinâmica de evolução é feita de forma adiabática, o Teorema Adiabático garante que a probabilidade de que o estado final do sistema seja o estado fundamental (desconhecido) de  $H_P$  é próxima de 1, encontrando assim a solução ótima do problema [60].

A adiabaticidade envolvida no processo de interpolação dos hamiltonianos  $H_I$  e  $H_P$  nos algoritmos de CQA não está relacionada com uma dinâmica onde não ocorrem trocas de calor como no *Classical Annealing*, mas com uma dinâmica de evolução quase estática onde não ocorrem transições entre os níveis de energia, garantindo que o estado que descreve o sistema seja aquele de menor energia em todos os instantes, conduzindo o sistema à solução ótima do problema de otimização ao final do processo [17]. A base para o funcionamento dos protocolos de CQA, portanto, advém do Teorema Adiabático.

Demonstrado pela primeira vez por Max Born e Vladimir Fock em 1928 [61], o Teorema Adiabático é um resultado fundamental da Mecânica Quântica que descreve a evolução de um sistema quântico quando uma perturbação externa suficientemente pequena é aplicada [62].

**(Teorema Adiabático).** *Um sistema físico permanece em seu autoestado instantâneo ao longo de uma evolução temporal se sobre ele é aplicada uma perturbação suficientemente pequena.*

Para que o Teorema Adiabático seja aplicável a um sistema exposto a uma perturbação externa, é necessário impor que a taxa de variação de  $H(s)$  seja ínfima quando comparada com o *gap* de energia entre o estado fundamental e o primeiro estado excitado instantâneos do operador [17], garantindo assim que não existem termos de mistura de estados durante o processo de interpolação e, portanto, que o estado que descreva o sistema em todos os instantes seja o estado fundamental. Condições de adiabaticidade

mais rigorosas podem ser encontradas nas Refs. [63, 64].

Em suma, a CQA é um modelo de Computação Quântica para abordar problemas de otimização. A abordagem se fundamenta no Teorema Adiabático para realizar a evolução partindo do estado fundamental do Hamiltoniano Inicial e chegando ao estado fundamental do Hamiltoniano Problema, no qual está codificada a melhor solução do problema. Para que a interpolação ocorra idealmente e a solução ótima seja encontrada, a taxa de variação da evolução de  $H(s)$  deve ser ínfima quando comparada com *gap* de energia entre o primeiro estado excitado e o estado fundamental instantâneos de  $H(s)$ .

Na realização prática desta abordagem em dispositivos quânticos reais, entretanto, existem diversas fontes de erro que não existem na formulação teórica da CQA. Nesse sentido, a qualidade dos qubits, o emaranhamento com o ambiente, erros de controle dos componentes que realizam fisicamente a interpolação entre os hamiltonianos e outras limitações físicas de cunho técnico e experimental [18, 19] podem afetar a efetividade da interpolação adiabática e permitir flutuações de energia durante a evolução, levando à soluções aproximadas, como mínimos locais, em detrimento da otimalidade da solução.

Em realidade, esta é a distinção entre CQA e *Quantum Annealing*. O QA é uma metaheurística que pode ser considerada a realização prática da CQA, estando portanto limitada aos erros que impedem que a interpolação adiabática seja feita idealmente, o que acarreta em possíveis transições durante a evolução do sistema, caindo em mínimos locais em vez da solução ótima.

Um efeito desses erros é que, na prática, nem sempre é possível ajustar  $A(1)$  e  $B(0)$  identicamente nulos na aplicação física da eq. (4.1). Nesses casos, é suficiente ajustar  $A(0) \gg B(0)$  e  $B(1) \gg A(1)$  - como mostra a Figura 1 - para que o sistema transite do estado fundamental inicialmente preparado para a solução ótima codificada no estado fundamental de  $H_P$ , ao menos aproximadamente [65].

Um outro apontamento importante acerca do QA é que, ainda que ocorram transições que levem a mínimos locais no decorrer do processo de interpolação, o tunelamento quântico induz uma probabilidade de que o sistema escape desse mínimo local para outras regiões mais promissoras do espaço de soluções. Essa probabilidade de tunelamento é mais eficiente do que os mecanismos usados por heurísticas clássicas que, *grosso modo*, precisam contornar as barreiras de potencial para escapar de mínimos locais. No *Classical Annealing*, por exemplo, a fuga desses mínimos, exige uma temperatura suficientemente alta para permitir a transição para outra região, resultando em tempos de resfriamento prolongado. Nesse sentido, estudos numéricos e experimentais corroboram a superioridade do QA sobre o *Classical Annealing* na exploração do espectro de um hamiltoniano [66].

Atualmente, algoritmos de CQA podem ser implementados nos computadores quânticos da empresa *D-Wave Systems*. que, por utilizarem a metaheurística de *Quantum*

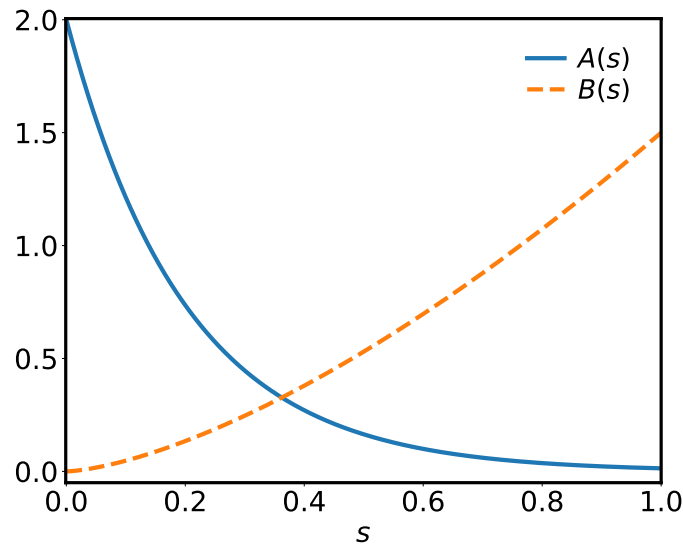


Figura 1 – Evolução temporal das funções de interpolação  $A(s)$  e  $B(s)$ . No gráfico, as condições de contorno  $A(0) \gg B(0)$  e  $B(1) \gg A(1)$  são genericamente satisfeitas com funções arbitrárias. Se a interpolação for feita de forma suficientemente lenta, o sistema é (ao menos, aproximadamente) levado ao estado fundamental de  $H_P$ , que codifica a solução ótima do problema.

*Annealing*, são conhecidos como *quantum annealers*. Desde 2017, o acesso a estes dispositivos pode ser feito via nuvem por diferentes plataformas. No Brasil, os *quantum annealers* da *D-Wave* podem ser acessados por nuvem (de forma paga) pela plataforma *Amazon Marketplace* [67], onde maiores informações sobre custos e configurações podem ser encontradas.

Na próxima seção, vamos nos aprofundar nos modelos mais avançados dos *quantum annealers* disponibilizados em nuvem para a implementação de algoritmos. Para isso, abordaremos aspectos que podem limitar o poder de resolução dessas implementações, como a quantidade de qubits e conectividade de cada topologia e o formato restrito em que os Hamiltonianos-Problemas devem ser formulados para realização da tarefa nesses dispositivos.

## 4.2 *D-Wave* e *Quantum Annealers*

Em 2007, a empresa canadense *D-Wave Systems* lançou o primeiro *quantum annealer* comercial, o *D-Wave One*, com 128 qubits [68]. Em 2011, o acesso ao *D-Wave One* foi disponibilizado na plataforma<sup>3</sup> *Leap*. Dois anos depois, o *D-Wave Two* foi lançado e também disponibilizado, contando com o dobro de qubits e diversas melhorias em relação ao seu antecessor [68].

<sup>3</sup> *Leap* [69] é uma plataforma da empresa *D-Wave Systems* que oferece acesso em nuvem a seus dispositivos. A plataforma só está disponível para acesso a partir de países da América do Norte, Europa e Ásia.

Atualmente, a empresa oferece acesso às versões mais recentes dos seus *quantum annealers* [70], incluindo o *D-Wave 2000Q*, com 2048 qubits, e o *D-Wave Advantage*, com 5640 qubits. O acesso aos dispositivos pode ser feito pelas plataformas *Leap* e *Amazon Marketplace*, sendo que somente a última pode ser acessada a partir da América do Sul.

Os qubits utilizados para a construção desses dispositivos são baseados em circuitos supercondutores de dimensões microscópicas que utilizam junções Josephson e operam em radiofrequência, conhecidos como CCJJ rf-SQUIDs (*compound-compound Josephson-junction radio frequency superconducting quantum interferences devices*) [71].

Uma junção Josephson é um dispositivo eletrônico que consiste em uma camada de material supercondutor separada por uma barreira isolante de uma camada de metal normal. Quando um potencial elétrico é aplicado à junção, um fluxo de elétrons através da barreira pode ocorrer por meio do fenômeno de tunelamento quântico. Esse fluxo de elétrons é altamente sensível a campos magnéticos, permitindo a utilização dessas junções como qubits supercondutores que podem ser manipulados para realizar cálculos quânticos [17].

Fisicamente, os qubits dos *quantum annealers* da *D-Wave* formam uma configuração equivalente a um modelo de Ising [17, 72], restringindo a capacidade de resolução de problemas desses dispositivos a hamiltonianos dependentes do tempo que podem ser escritos como [73]

$$H(s) = A(s) \sum_i \sigma_x^i + B(s) \left( \sum_i h_i \sigma_z^i + \sum_{\substack{i,j \\ i>j}} J_{ij} \sigma_z^i \sigma_z^j \right), \quad (4.2)$$

onde os operadores  $\sigma_\alpha^i$  ( $\alpha = x, z$ ) correspondem às matrizes de Pauli para o  $i$ -ésimo spin,  $h_i$  é a anisotropia aplicada ao  $i$ -ésimo spin e  $J_{ij}$  é o acoplamento entre o  $i$ -ésimo e  $j$ -ésimo spins.

No sistema, em conexão com a Figura 4.1, o processo de interpolação expresso pela eq. (4.2) é realizado partindo de uma configuração na qual os spins estão axialmente alinhados com campos magnéticos aplicados que então são lentamente reduzidos ao mesmo passo em que as anisotropias e os acoplamentos entre os qubits crescem na direção  $z$  [17], idealmente levando ao estado fundamental do hamiltoniano que acompanha  $B(s)$ .

Dessa forma, comparando as equações (4.1) e (4.2), conclui-se que, para que seja possível implementar um problema de otimização nos *quantum annealers* da *D-Wave*, o Hamiltoniano Problema deve ser escrito na forma

$$H_P = \sum_i h_i x_i + \sum_{i>j} J_{ij} x_i x_j, \quad (4.3)$$

onde introduzimos as variáveis binárias  $x_{i,j} = (1 + \sigma_z^{i,j})/2$ , que assumem os valores 0 ou 1. Este formato restrito de hamiltoniano é conhecido como QUBO [74].

Os hamiltonianos do tipo QUBO (*Quadratic Unconstrained Binary Optimization*<sup>4</sup>) formam uma classe de operadores particularmente úteis para abordar problemas de otimização combinatória que possam ser formulados com o uso de variáveis binárias [75]. A formulação matemática desses hamiltonianos envolve a soma de dois termos: um termo linear e um termo quadrático-cruzado nas variáveis binárias. O termo linear representa a contribuição individual de cada variável e o termo quadrático representa a interação (ou o acoplamento) entre as variáveis.

Ao formular um problema de otimização como um hamiltoniano desta classe, é necessário incluir todas as restrições do problema já na formulação matemática do hamiltoniano. Essas restrições podem incluir limitações na combinação de variáveis, limitações no número de variáveis que podem ser definidas como 1, ou qualquer outra condição que seja necessária para impôr as exigências da tarefa em questão. Ao fazer isso, dizemos que a formulação do problema é irrestrita, pois todas as restrições já estão incluídas em sua própria formulação.

Além da limitação relacionada à formulação matemática do Hamiltoniano Problema em um hamiltoniano do tipo QUBO, existe outra característica importante de ser levada em conta ao utilizar os *quantum annealers* da *D-Wave*. Devido a limitações da engenharia desses processadores, os dispositivos não possuem interconectividade total e os acoplamentos entre os qubits são definidos pelas diferentes topologias. A Figura 2a é uma representação da topologia *Chimera*, presente nos *quantum annealers D-Wave 2000Q*, que possuem 2048 qubits conectados seis a seis [76]. Já a Figura 2b, por sua vez, representa a topologia *Pegasus*, presente nos *quantum annealers Advantage*, que possuem 5640 qubits conectados quinze a quinze [76].

No entanto, na abordagem de problemas de otimização nos computadores quânticos da *D-Wave*, é comum que o número de conexões entre as variáveis lógicas necessárias extrapole o número de conexões entre os qubits da topologia que se adota. Nesse caso, para contornar a falta de conexão entre os qubits, os dispositivos oferecem uma ferramenta conhecida como *minor embedding* [77], que mapeia o número de conexões existentes no Hamiltoniano Problema em uma determinada seção da topologia de destino, utilizando qubits auxiliares para efetivamente conectar diferentes células da topologia e aumentar o número de conexões entre os qubits. Essa ferramenta, entretanto, aumenta significativamente o número de qubits físicos utilizados para implementação de um dado Hamiltoniano Problema quando comparado com o número de qubits lógicos necessários na teoria.

Nesta seção, apresentamos aspectos gerais dos *quantum annealers* da *D-Wave*, como uma breve descrição da sua engenharia e a restrição acerca do tipo de Hamiltoniano Problema que pode ser implementado. Discutimos também aspectos mais específicos das topologias dos *quantum annealers*, como a quantidade de qubits, a conectividade apenas

<sup>4</sup> Otimização Binária Quadrática Irrestrita, em tradução livre.

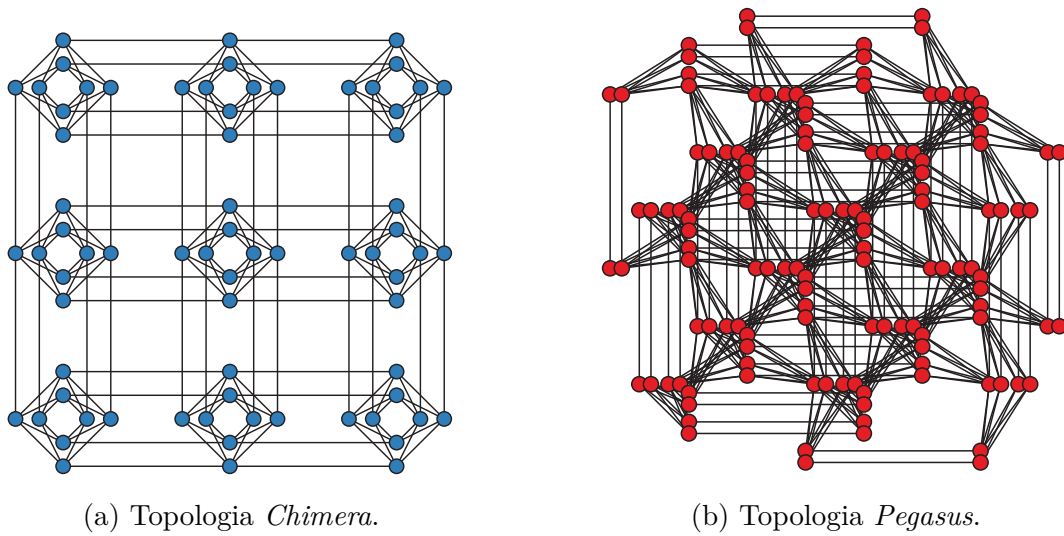


Figura 2 – Representação das topologias presentes nos *quantum annealers* da *D-Wave*. Em (a), a topologia *Chimera*, presente nos chips *D-Wave 2000Q*. Em (b), a topologia *Pegasus*, presente nos chips *Advantage*. Extraído de [17].

parcial entre eles e a técnica de *minor embedding*, usada para contornar estes obstáculos ao custo de um aumento do número de qubits físicos utilizados.

Nas próximas seções, vamos demonstrar como é realizada a construção dos Hamiltonianos Problemas utilizados para implementar algoritmos que abordam problemas de otimização nos *quantum annealers* da *D-Wave*. Como exemplo, vamos construir um Hamiltoniano Problema para o Problema do Caixeiro Viajante e outro para o Problema da Mochila, discutindo as características dessas implementações.

### 4.3 Quantum Annealing para o Problema do Caixeiro Viajante

Para o Problema do Caixeiro Viajante, definido na Seção 3.1, vamos construir um Hamiltoniano Problema da forma [28]

$$H_{TSP} = H_A + H_B + H_C, \quad (4.4)$$

onde cada autoestado codifica uma diferente sequência de cidades (isto é, uma rota) que o caixeiro deve visitar, associando a cada sequência uma autoenergia. Com a ideia de utilizar o Teorema Adiabático e encontrar a solução ótima (ou de alta qualidade, ao menos), devemos codificar os autoestados de tal forma que quão maior a qualidade da solução, menor a energia associada.

Para isso, os termos de  $H_{TSP}$  devem introduzir (i) as restrições presentes no Problema do Caixeiro Viajante e (ii) a função a ser minimizada. Dessa forma, os termos devem ser formulados de tal forma que (a) as soluções que ferem as restrições impostas sejam penalizadas com altas energias (proporcionais à gravidade com a qual as restrições

foram feridas) e (b) as soluções que respeitam as restrições não sejam penalizadas e compitam entre si apenas sendo influenciadas pelo termo que introduz a função a ser minimizada. Desse modo, quão melhor a solução, menor será energia a ela associada, justamente como apontado.

Assim, queremos em primeiro lugar construir um hamiltoniano que codifique a função a ser minimizada, isto é, um hamiltoniano cujo espectro esteja associado às distâncias que o caixeiro percorre nas viagens entre as  $N$  cidades de uma rota. Para isso, vamos definir a variável binária  $x_{i,l}$  tal que

$$x_{i,l} = \begin{cases} 1, & \text{caso o caixeiro esteja na } i\text{-ésima cidade no intervalo de tempo } l, \\ 0, & \text{caso contrário,} \end{cases} \quad (4.5)$$

onde ( $i = 1, \dots, N$ ) e ( $l = 1, \dots, N$ ), e consideremos o hamiltoniano

$$H_A = \sum_{\substack{i,j=1 \\ i>j}}^N W_{ij} \sum_{l=1}^N x_{i,l} x_{j,l+1}, \quad (4.6)$$

que codifica as distâncias  $W_{ij}$  que o caixeiro viaja nos  $N$  intervalos de tempo necessários para viajar entre as  $N$  cidades de uma rota.

Em segundo lugar, é necessário introduzir hamiltonianos que penalizam as soluções que ferem as restrições do problema. Para isso, consideremos o hamiltoniano

$$H_B = \lambda_1 \sum_{i=1}^N \left( 1 - \sum_{l=1}^N x_{i,l} \right)^2, \quad (4.7)$$

que introduz uma penalização energética proporcional à  $\lambda_1$  às soluções em que o caixeiro visita uma mesma cidade múltiplas vezes, e o hamiltoniano

$$H_C = \lambda_2 \sum_{l=1}^N \left( 1 - \sum_{i=1}^N x_{i,l} \right)^2, \quad (4.8)$$

que introduz uma penalização energética proporcional à  $\lambda_2$  às soluções em que o caixeiro está em mais que uma (ou nenhuma) cidade durante o mesmo intervalo de tempo  $l$ .

Os parênteses das eqs. (4.7) e (4.8) são elevados ao quadrado a fim de garantir que as penalizações energéticas introduzidas sejam sempre positivas e, portanto, que os estados que codificam as soluções que ferem as restrições do problema estejam distantes da solução ótima e de sua vizinhança, na qual estão codificadas soluções aproximadas de alta qualidade. Além disso, é importante ressaltar que é a necessidade de introduzir às restrições citadas que justifica a introdução da variável  $l$  na variável binária do problema.

Assim, o hamiltoniano total é dado por

$$H_{TSP} = \sum_{\substack{i,j=1 \\ i>j}}^N W_{ij} \sum_{l=1}^N x_{i,l} x_{j,l+1} + \lambda \left[ \sum_{i=1}^N \left( 1 - \sum_{l=1}^N x_{i,l} \right)^2 + \sum_{l=1}^N \left( 1 - \sum_{i=1}^N x_{i,l} \right)^2 \right], \quad (4.9)$$

onde tomamos  $\lambda = \lambda_1 = \lambda_2$  por conveniência. Notemos que o termo entre colchetes, isto é, o termo de penalização, só é nulo quando todas as condições do TSP são satisfeitas.

A respeito do parâmetro  $\lambda$ , é importante considerar que seu valor numérico deve ser ajustado de tal forma que seja suficientemente grande para que as condições impostas no hamiltoniano de penalização não sejam violadas<sup>5</sup>. Caso  $\lambda$  não seja escolhido propriamente, é possível que o algoritmo retorne soluções que não sejam aceitáveis, como rotas que não cobrem todas as cidades. Nesse sentido, um ponto de partida no ajuste do parâmetro  $\lambda$  é fazer  $\lambda > \max W_{ij}$ .

Agora, antes de prosseguirmos com os resultados das implementações do Problema do Caixeiro Viajante nos *quantum annealers* da *D-Wave*, vale notar que  $H_{TSP}$  é, de fato, um problema do tipo QUBO, o que torna a implementação possível. Além disso, a variável binária que introduzimos,  $x_{i,l}$ , pressupõe que o caixeiro leva  $N$  intervalos de tempos para visitar  $N$  cidades. Logo,  $N^2$  qubits lógicos são necessários para a implementação do problema.

Assim, com o objetivo de validar o funcionamento do algoritmo quântico apresentado nesta seção, implementamos o hamiltoniano introduzido na eq. (4.9) no *quantum annealer Advantage 4.1* da *D-Wave* para instâncias com seis e sete cidades. Nesses casos, o algoritmo encontrou a menor rota possível, como mostra a Figura 3 para instâncias selecionadas.

Para que isso fosse possível, no caso com seis (sete) cidades, foram usados 140 (254) qubits físicos para mapear os 36 (49) qubits lógicos necessários. Além disso, para garantir que o estado fundamental fosse medido pelo menos uma vez, foram realizados 500 (1000) *shots*. Cada *shot* levou o tempo característico do processo de *annealing* de 18 microssegundos. Os resultados se mostraram exatos quando comparados com o Método da Força Bruta, cuja execução levou, em computador pessoal, um milissegundo (sete milissegundos).

## 4.4 Quantum Annealing para o Problema da Mochila

Para o Problema da Mochila, definido na Seção 3.2, vamos construir um Hamiltoniano Problema da forma

$$H_{KP} = H_A + H_B + H_C, \quad (4.10)$$

<sup>5</sup> É apropriado discutir com mais cuidado a necessidade de ajustar os parâmetros corretamente. Por exemplo, se  $\lambda$  não for suficientemente alto, soluções em que o caixeiro viajante deixa de passar em uma ou mais cidades podem corresponder à rotas com energias menores do que aquelas nas quais o caixeiro passa por todas as cidades. Isso acontece porque, dependendo dos valores dos parâmetros, é energeticamente mais barato receber uma punição proporcional à  $\lambda$  do que somar a distância entre as cidades ignoradas na rota. Nesse sentido, dizer que existe um ajuste correto entre os parâmetros é análogo a dizer que existe uma relação de desigualdade entre as variáveis  $W_{ij}$  e  $\lambda$  que deve ser respeitada para que o processo de *annealing* encontre soluções que respeitam às restrições do problema.

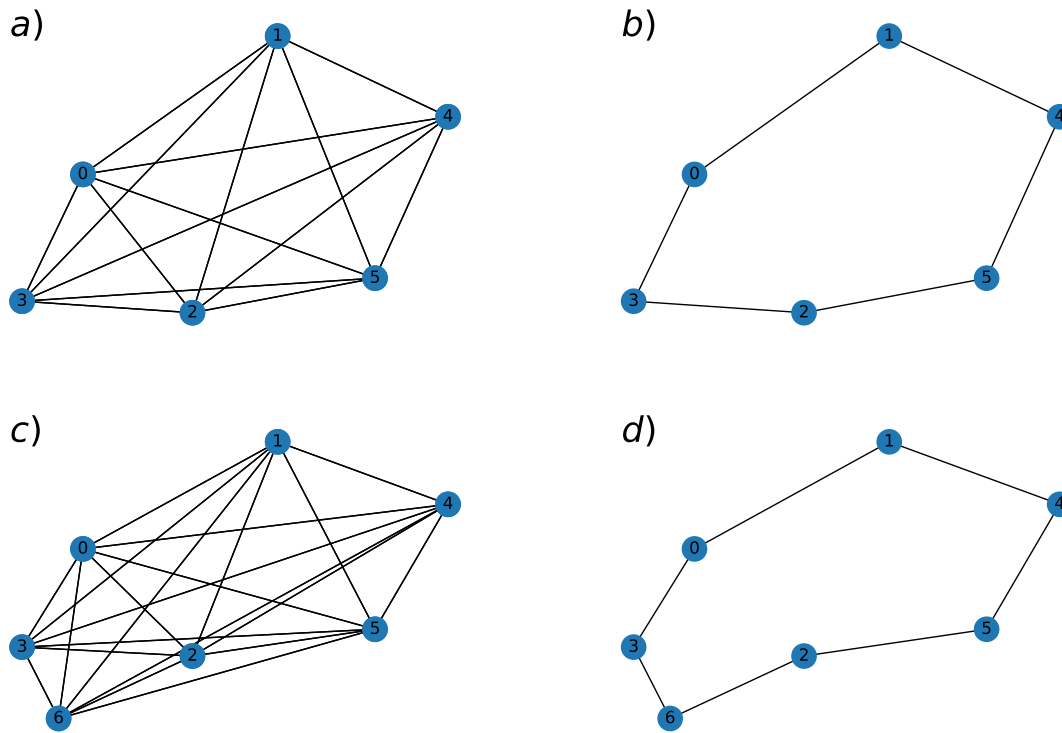


Figura 3 – Representação do Problema do Caixeiro Viajante para 6 e 7 cidades. A Figura a) mostra todas as possíveis conexões para o caso de 6 cidades distribuídas aleatoriamente no espaço e a Figura b) mostra a solução ótima encontrada no *quantum annealer*. A Figura c) mostra todas as possíveis conexões para o caso de 7 cidades distribuídas aleatoriamente no espaço e a Figura d) mostra a solução ótima encontrada no *quantum annealer*.

onde cada autoestado codifica uma possível configuração da mochila, associando a cada configuração uma autoenergia. Novamente, com a ideia de utilizar o Teorema Adiabático e encontrar a solução ótima (ou de alta qualidade, ao menos), devemos codificar os autoestados de tal forma que quão maior a qualidade da solução, menor a energia associada.

Para isso, os termos de  $H_{KP}$  devem introduzir (i) restrições que garantam que a soma dos pesos  $w_\alpha$  dos itens contidos na mochila seja menor ou igual que o limite  $L$  e (ii) a função que maximiza o valor total da mochila. Aqui, vamos nos restringir ao caso em que  $w_\alpha$  e  $L \in \mathbb{N}$ .

Em primeiro lugar, queremos construir um hamiltoniano que reforce que a mochila pode ter apenas um peso final, que vamos definir como  $n$ ,  $n \leq L$ . Para isso, vamos definir a variável binária  $y_n$ , onde  $(n = 1, \dots, L)$ , tal que

$$y_n = \begin{cases} 1, & \text{se o peso final da mochila é } n, \\ 0, & \text{caso contrário.} \end{cases} \quad (4.11)$$

e consideremos o hamiltoniano

$$H_A = \lambda_1 \left( 1 - \sum_{n=1}^L y_n \right)^2, \quad (4.12)$$

que introduz uma penalização energética proporcional à  $\lambda_1$  às soluções em que a mochila tem mais que um (ou nenhum) peso final.

Em segundo lugar, queremos impôr que o peso final da mochila deve ser igual à soma dos pesos dos itens nela contidos. Para isso, introduzimos a variável binária  $x_\alpha$ , onde ( $\alpha = 1, \dots, N$ ), tal que

$$x_\alpha = \begin{cases} 1, & \text{caso o } \alpha\text{-ésimo item esteja contido na mochila,} \\ 0, & \text{caso contrário.} \end{cases} \quad (4.13)$$

Dessa forma, podemos considerar o hamiltoniano

$$H_B = \lambda_2 \left( \sum_{n=1}^L n y_n - \sum_{\alpha=1}^N w_\alpha x_\alpha \right)^2, \quad (4.14)$$

que introduz uma penalização energética proporcional à  $\lambda_2$  às soluções em que o peso final da mochila (primeiro somatório) é diferente da soma dos itens nela contido (segundo somatório).

E, por fim, queremos construir um hamiltoniano que codifique a função a ser maximizada, isto é, cujo espectro esteja associado ao valor dos itens adicionados na mochila. Para isso, consideremos o hamiltoniano

$$H_C = -\gamma \sum_{\alpha=1}^N v_\alpha x_\alpha, \quad (4.15)$$

que codifica o valor  $v_\alpha$  de cada item na mochila de tal forma que quão mais caros são os itens carregados, menor é a energia do hamiltoniano total. O parâmetro  $\gamma$  deve ser ajustado em relação à  $\lambda_1$  e  $\lambda_2$ .

Assim, o hamiltoniano total é dado por

$$H_{KP} = \lambda \left[ \left( 1 - \sum_{n=1}^L y_n \right)^2 + \left( \sum_{n=1}^L n y_n - \sum_{\alpha=1}^N w_\alpha x_\alpha \right)^2 \right] - \gamma \sum_{\alpha=1}^N v_\alpha x_\alpha, \quad (4.16)$$

onde tomamos  $\lambda = \lambda_1 = \lambda_2$  por conveniência. Notemos que o termo entre colchetes, isto é, o termo de penalização, só é nulo quando todas as condições do KP são satisfeitas.

A respeito do balanceamento dos parâmetros, os parâmetros devem ser ajustados de tal forma que as condições impostas pelo termo de penalização não sejam violadas às custas de adicionar mais itens à mochila, tornando  $H_C$  mais negativo. Nesse sentido, um ponto de partida no ajuste dos parâmetros é  $0 < \gamma \max(v_\alpha) < \lambda$ .

Novamente, vale notar que a eq. (4.16) é, de fato, um problema do tipo QUBO, o que torna possível a implementação do Problema da Mochila como acima formulado nos *quantum annealers* da *D-Wave*. Notemos, ainda, que introduzimos  $L$  variáveis binárias  $y_n$  e  $N$  variáveis binárias  $x_\alpha$ . Logo,  $N + L$  qubits lógicos são necessários para a implementação do problema.

Agora, com o intuito de reduzir o número de qubits necessários na implementação do Problema da Mochila, consideremos uma reformulação do hamiltoniano apresentado na eq. (4.16), dada por

$$H_{KP} = \lambda \left( L - \langle \mathbf{2} | \mathbf{a} \rangle - \sum_{\alpha=1}^N w_\alpha x_\alpha \right)^2 - \gamma \sum_{\alpha=1}^N v_\alpha x_\alpha, \quad (4.17)$$

onde adotamos a notação

$$\langle \mathbf{2} | \mathbf{a} \rangle = \sum_{l=0}^{\Omega} 2^l a_l, \quad (4.18)$$

com  $\Omega = \lfloor \log_2 L \rfloor$ , para representar a expansão binária sobre as novas variáveis auxiliares<sup>6</sup>  $a_l$ .

Nesta nova formulação, o primeiro termo introduz a restrição característica de que a soma do peso dos itens contidos na mochila não deve exceder sua capacidade máxima. Para que isso seja possível, introduzimos a expansão binária  $\langle \mathbf{2} | \mathbf{a} \rangle$ , que pode assumir qualquer valor entre 0 e  $L$ , garantindo que o termo dentro dos parênteses seja nulo desde que a soma do peso dos itens seja menor ou igual a  $L$ . Uma penalização energética proporcional à  $\lambda$  é introduzida às soluções que violam essa restrição.

Vale ressaltar que para realizar a expansão binária e representar os  $L + 1$  números inteiros que existem entre 0 e  $L$ , apenas  $\lfloor \log_2 L \rfloor + 1$  variáveis binárias são necessárias. Por exemplo, o número 5 pode ser representado pelas variáveis  $a_0 = 1$ ,  $a_1 = 0$  e  $a_2 = 2$ . Dessa forma, o número total de qubits lógicos necessários nesta nova formulação é igual a  $N + \lfloor \log_2 L \rfloor + 1$  em comparação com os  $N + L$  qubits necessários na formulação anterior. Na literatura, a redução do número de variáveis binárias necessárias para codificar um problema de otimização usando uma expansão binária é conhecida como *log trick* [28].

## 4.5 Breve Discussão sobre o *Quantum Annealing* para o Problema do Roteamento de Veículos

Para que um problema de otimização possa ser implementado nos *quantum annealers* da *D-Wave*, deve ser possível formulá-lo como um hamiltoniano do tipo QUBO. Entre

<sup>6</sup> Em inglês, estas variáveis são chamadas de *slack variables* (variáveis de folga ou variáveis soltas, em tradução literal). Aqui, adotamos a nomenclatura menos corajosa de variáveis auxiliares. Essas variáveis são utilizadas na formulação dos hamiltonianos QUBOs e servem para flexibilizar a forma com que as restrições são impostas.

muitos outros [28], o TSP e o KP são exemplos de problemas de otimização que já foram formulados nesse formato específico e, portanto, podem ser implementados nesses dispositivos, como visto nas seções anteriores.

Considerando que as tarefas supracitadas estão por trás de uma grande variedade de logísticas, muitos trabalhos têm se dedicado a explorar o potencial da implementação dessas tarefas na abordagem de aplicações de problemas práticos, como o Problema do Roteamento de Veículos.

Na Ref. [24], um grupo de pesquisa da Universidade Ludwig Maximilian de Munique, em parceria com a fabricante de veículos alemã *Volkswagen*, utilizou os *quantum annealers* da *D-Wave* para abordar o Problema do Roteamento de Veículos com Capacidade, como definido na Seção 3.3. Para isso, a tarefa foi dividida em duas fases: a Fase de Clusterização (mapeada em um Problema de Múltiplas Mochilas) e a Fase de Roteamento (mapeada em um Problema do Caixeiro Viajante), como também descrito na Seção 3.3.

Na referência, os autores desenvolveram três algoritmos: (i) um algoritmo quântico em que as fases são abordadas em sequência, (ii) um algoritmo quântico em que as fases são abordadas simultaneamente e (iii) um algoritmo híbrido em que a Fase de Clusterização é abordada classicamente e a Fase de Roteamento é abordada quanticamente.

Os resultados comparativos mostram que, entre os três algoritmos desenvolvidos, o algoritmo híbrido se mostrou o mais eficaz, competindo em termos de qualidade das soluções (mas não em termos de tempo de execução) com heurísticas já consolidadas para a tarefa. Apontam ainda que as limitações em relação ao número de qubits e a baixa conectividade entre eles deixam questões em aberto sobre o real ganho das estratégias desenvolvidas.

Ainda, na Ref. [56], um grupo de pesquisa da fabricante de veículos japonesa *Toyota* utilizou os *quantum annealers* da *D-Wave* para implementar uma formulação do CVRP equipado a janelas de tempo que descrevem a evolução da posição e da capacidade de cada veículo. Segundo os autores, os resultados obtidos atestam o funcionamento do algoritmo para instâncias com até 7 clientes. No entanto, também afirmam que, para aplicação em casos do mundo real, são necessários pelo menos 2000 qubits lógicos.

## 4.6 *Quantum Annealing* para o Problema de Gerenciamento de Estoque

Até aqui, discutimos como problemas de otimização combinatória podem ser encontrados sob os mais diferentes contextos. Discutimos também como o crescimento explosivo do espaço de soluções dessas tarefas é um impeditivo para a computação clássica no desenvolvimento de algoritmos que sejam simultaneamente eficientes e exatos,

justificando assim o crescente interesse e investimento de diversos setores da indústria em novas abordagens para esses problemas, como a Computação Quântica Adiabática.

Em específico, nas seções anteriores, detalhamos como o Problema do Caixeiro Viajante e o Problema da Mochila, tarefas amplamente utilizadas na modelagem de uma grande variedade de tarefas de interesse comercial, já foram abordados em algoritmos desenvolvidos nesse modelo de Computação Quântica. Ainda, exploramos os *quantum annealers* da *D-Wave*, dispositivos apropriados para a implementação desses algoritmos, recebendo esse nome por utilizarem a meta-heurística de *Quantum Annealing* para realizar o processo físico necessário para realização de CQA.

Seguindo os objetivos dessa dissertação, dedicamos o próximo capítulo a apresentar o Problema do Gerenciamento de Estoque, um problema de otimização combinatória ligado à eficiência organizacional de depósitos e centros de distribuição.

## 5 Problema de Gerenciamento de Estoque

Dentro de uma fábrica, o estoque representa o conjunto dos insumos necessários para abastecer outros setores, como as linhas de produção. A organização eficiente do depósito, onde o estoque é armazenado, torna-se assim uma necessidade complexa, sobretudo devido ao fluxo constante de itens necessários para atender às demandas operacionais [78].

Desse modo, considerando a importância dessa organização para o funcionamento eficiente de uma fábrica, determinar a forma mais vantajosa de distribuir os itens do estoque nos sistemas de armazenamento do depósito (isto é, a melhor combinação) resulta na redução do número de operações necessárias nos processos de retirada de itens, que por sua vez implica na diminuição da utilização de equipamentos de movimentação de carga<sup>1</sup> e na prevenção de acidentes no ambiente fabril.

Problemas de otimização ligados à eficiência organizacional do depósito de uma fábrica recebem o nome de Problema de Gerenciamento de Estoque. O objetivo central dessas tarefas é determinar a alocação ideal para os itens do estoque nos sistemas de armazenamento do depósito, minimizando os custos operacionais associados à logística de entrada e saída de insumos [25, 79].

Como existem diferentes modelos de fábricas com estruturas físicas e necessidades organizacionais diversas [25], uma estratégia que se proponha a otimizar o gerenciamento do estoque de uma fábrica deve considerar informações específicas sobre ela, que incluem:

- Unidade padrão de armazenamento adotada: É essencial conhecer a unidade padrão de armazenamento adotada. Em fábricas de médio e grande porte, por exemplo, é comum a utilização de caixas-paletes<sup>2</sup> (ou, abreviadamente, paletes) no armazenamento dos insumos, como ilustrado na Figura 4.
- Sistema de armazenamento adotado: É necessário também conhecer o sistema de armazenamento adotado pela fábrica. Em fábricas de médio e grande porte, nas quais paletes são utilizados como unidade padrão de armazenamento, a utilização de prateleiras gravitacionais é muito comum.

Prateleiras gravitacionais são sistemas projetados para permitir a movimentação automática dos paletes através de trilhos levemente inclinados [26], que permitem o deslizamento por ação da força gravitacional, como ilustrado na Figura 5.

<sup>1</sup> Equipamentos de movimentação de carga são máquinas empregadas no deslocamento de itens em estoques, como empilhadeiras, transpaletes e transelevadores.

<sup>2</sup> Caixa-paleta é um recipiente de armazenamento e transporte que é construído sobre uma base de paleta, permitindo a movimentação por meio de equipamentos de movimentação de cargas.



Figura 4 – Representação de uma caixa-paleta. Em fábricas de médio e grande porte, caixas-paletes são utilizadas como unidade padrão de armazenamento. O conteúdo dessas caixas-paletes são insumos requeridos pelas linhas de produção na montagem dos produtos finais. Extraído de [80].



Figura 5 – Representação de um conjunto de prateleiras gravitacionais. Em fábricas de médio e grande porte, prateleiras gravitacionais são adotadas para armazenar os paletes no depósito. Na ilustração, a empilhadeira do lado esquerdo introduz uma caixa-paleta em uma prateleira gravitacional (trilho), que deslizará até encostar nos demais paletes ali alocados, enquanto a empilhadeira do lado direito faz a retirada de um outro paleta requisitado pela linha de produção. Extraído de [81].

Essas prateleiras seguem uma lógica conhecida como FIFO (*First In, First Out*) [27], previamente definida, em que a ordem de retirada dos itens segue a ordem de inserção, ou seja, os paletes são retirados na ordem em que foram inseridos, evitando a deterioração dos insumos.

Dessa forma, quando um paleta alocado em uma prateleira gravitacional é requisitado, é necessário remover todos os paletes que foram inseridos antes dele. Esses paletes retirados devem ser reinserto no depósito. Por sua vez, a reinserto de um paleta não precisa necessariamente ocorrer na mesma prateleira em que ele estava originalmente alocado.

- Informações sobre os itens: Informações sobre o conteúdo, as dimensões, o peso e o horário de entrada de cada paleta no depósito são essenciais para o gerenciamento do estoque.

Na prática, essas e outras informações relevantes para a identificação, rastreamento

e gerenciamento de um palete são armazenadas em um código alfanumérico único denominado SKUID (*Stock Keeping Unit Identifier*). Dessa forma, dentro de uma fábrica, cada palete (ou SKU) possui seu próprio SKUID, sendo esta a nomenclatura padrão adotada no ambiente fabril para se referir a um determinado insumo armazenado no estoque.

- Padrão de ocorrência entre os itens: Uma informação fundamental para determinar a alocação ideal dos paletes nas prateleiras é o padrão de ocorrência conjunta dos insumos nas demandas, isto é, quão frequentemente certos insumos são solicitados em conjunto. A análise desses dados permite determinar quais paletes devem preferencialmente ser alocados nas mesmas prateleiras em detrimento de outros.

Por exemplo, em uma fábrica de automotivos, durante o processo de montagem, as maçanetas devem ser requisitadas juntamente com as portas, mas não necessariamente com os bancos. Nesse sentido, entre portas e bancos, é preferível que as maçanetas estejam alocadas na mesmas prateleiras que as portas, embora todos os itens sejam usados na montagem do produto final.

Portanto, o padrão de ocorrência conjunta entre os itens é um indicativo confiável para dizer quais insumos devem ser preferencialmente alocados em conjunto.

- Política organizacional: As restrições impostas pela política organizacional da fábrica também devem ser levadas em conta. Isso pode incluir preferências ou requisitos específicos sobre a organização dos insumos no estoque, como agrupar ou separar paletes com mesmo conteúdo.

## 5.1 Um Exemplo Real

Nesta seção, temos como objetivo aprofundar nosso entendimento sobre o funcionamento de um depósito. Para isso, obtivemos acesso às estruturas e aos dados da fábrica que é parceira desse projeto e analisamos os detalhes relacionados ao armazenamento de estoque, os quais resumidamente apresentaremos aqui. Em respeito ao acordo de confidencialidade que permite o acesso a essas informações sensíveis, omitiremos não apenas o nome da fábrica, mas também qualquer outra informação que possa levar a sua identificação.

No depósito da fábrica em questão, os insumos são armazenados em três modelos de paletes com dimensões geométricas distintas, que são alocados em prateleiras gravitacionais também de três modelos diferentes. As dimensões desses paletes e prateleiras são ilustradas nas Figuras 6 e 7, respectivamente, oferecendo uma visualização clara de cada componente do depósito.

Uma análise cuidadosa das dimensões indicadas nas Figuras 6 e 7 pode suscitar questionamentos em relação a quais tipos de paletes podem ser armazenados em cada

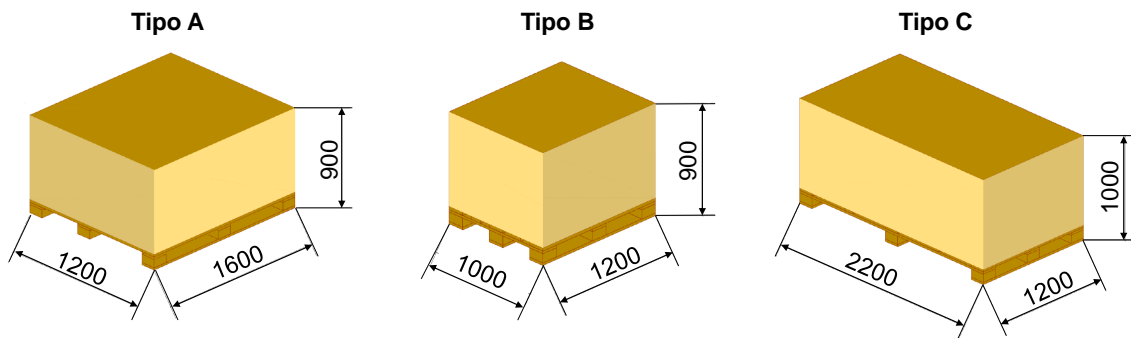


Figura 6 – As dimensões (em milímetros) dos três tipos de paletes que podem ser armazenados no depósito da fábrica são apresentadas. A largura é representada como a direção de encaixe da empilhadeira.

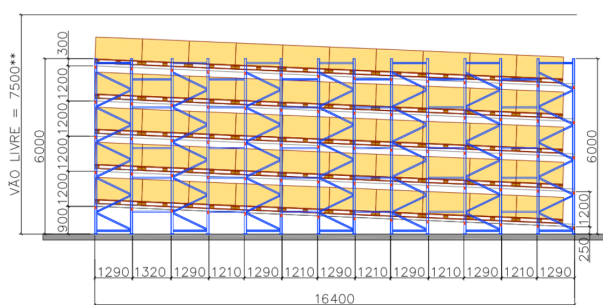
modelo de prateleira. Em relação a isso, a restrição de que os paletes devem deslizar sem girar é determinante. Portanto, as seguintes informações devem ser consideradas:

- Paletes do tipo A devem ser armazenados em prateleiras do Modelo A,
- Paletes do tipo B podem ser armazenados em prateleiras dos Modelos A e B, mas devem ser preferencialmente alocados no segundo modelo, de acordo com a política organizacional do depósito, e
- Paletes do tipo C devem ser armazenados em prateleiras do Modelo C.

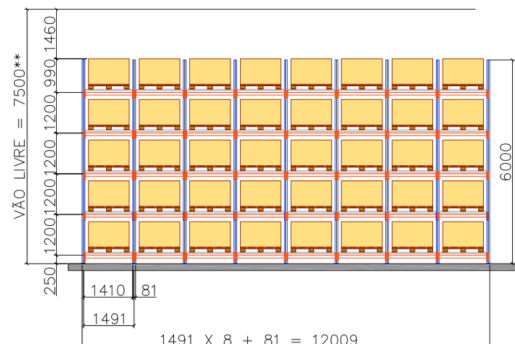
Em relação à quantidade de componentes no depósito, atualmente cerca de 4040 paletes (que armazenam 144 tipos de insumos) estão distribuídos em 205 prateleiras gravitacionais do Modelo A, 340 prateleiras gravitacionais do Modelo B e 20 prateleiras gravitacionais do Modelo C. A configuração representa uma ocupação de aproximadamente 60% da capacidade total do depósito, que pode alocar até 6730 paletes. Em média, 1200 paletes entram/saem do depósito todos os dias.

Uma vez que o sistema de armazenamento adotado pelo depósito da fábrica faz uso de prateleiras gravitacionais, a otimização do gerenciamento do estoque deve ser alcançada através da otimização das operações de alocação de paletes nessas prateleiras. Sendo assim, os processos a serem otimizados são dois: (i) o processo de inserção, quando um paleta é recebido pela fábrica e inserido no depósito pela primeira vez, e (ii) o processo de reinsertão, no qual, seguindo o princípio de gerenciamento FIFO, um paleta é retirado de uma prateleira para permitir a entrega de um segundo paleta à linha de produção, exigindo que o primeiro paleta seja reinsertido no depósito, podendo ser alocado em uma prateleira diferente da original. Como a base de ambos os processos é a mesma, isto é, determinar em qual prateleira um paleta deve ser inserido, a otimização do gerenciamento do estoque se reduz a tarefa de otimizar o processo de inserção.

**Modelo A**

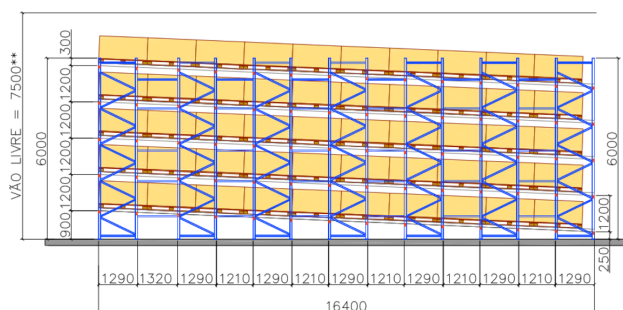


Vista Lateral

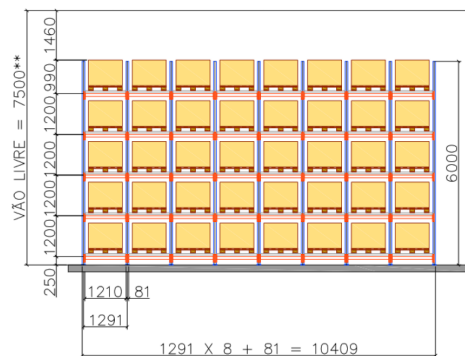


Vista Frontal

**Modelo B**

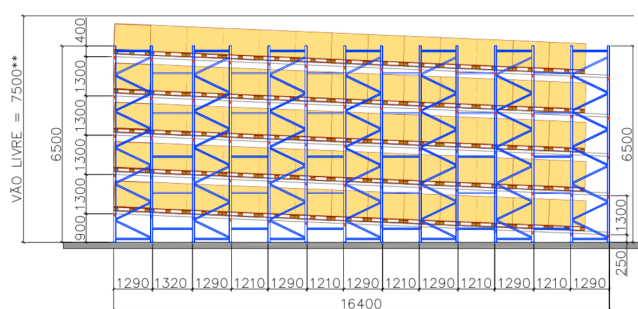


Vista Lateral

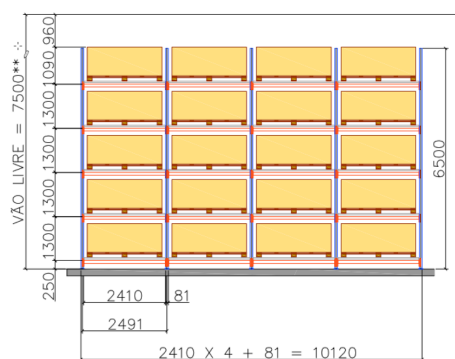


Vista Frontal

**Modelo C**



Vista Lateral



Vista Frontal

Figura 7 – As dimensões (em milímetros) dos três modelos de prateleiras disponíveis no estoque são ilustradas. Nos três modelos, a vista lateral apresenta uma plataforma com 5 prateleiras gravitacionais, enquanto a vista frontal mostra um conjunto dessas plataformas, onde apenas o primeiro palete é visível e pode ser removido pela empilhadeira.

Em outras palavras, para otimizar o gerenciamento do estoque no modelo da fábrica analisada, que utiliza prateleiras gravitacionais como sistema padrão de armazenamento em qual prateleira gravitacional cada paleta a ser inserido (ou reinserido) deve ser alocado para que o número de operações necessárias na entrega de futuras demandas (isto é, o número de reinserções) seja minimizado.

Nesse sentido, é importante registrar que, no momento de início desse projeto, a fábrica de nossa análise não possuía qualquer estratégia de gerenciamento de estoque. Em verdade, os processos de alocação ocorriam baseados no princípio de que paletes de mesmo conteúdo deveriam ser alocados nas mesmas prateleiras, negligenciando a correspondência entre os insumos, e dedicando, para cada insumo, uma prateleira inteira. Assim, a ausência de uma estratégia de gerenciamento do estoque resultava em uma organização ineficiente do estoque, gerando atrasos nas linhas de produção, além do uso ineficiente dos sistemas de armazenamento da fábrica, sendo este o principal problema, conforme relatado.

Como consequência desses problemas, a ineficiência na gestão do depósito gerada pela falta de uma estratégia de gerenciamento de estoque havia se tornado um impeditivo nos planos de expansão de vendas da fábrica, uma vez que, com o aumento do número de clientes, a demanda sobre o depósito também aumentaria, e já não existiam prateleiras disponíveis para que seguissem a organização no depósito como era feita.

Ainda, como possível justificativa para a ausência de uma estratégia na fábrica de nossa análise, durante a fase de revisão na literatura da área, não encontramos trabalhos que se dedicassem à otimização do gerenciamento de estoque voltado para fábricas que utilizam prateleiras gravitacionais. Em verdade, a literatura existente [78, 79, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91] trata, em sua maioria, da otimização de gerenciamento de estoque sobre outros sistemas de armazenamento, como prateleiras de distribuição multinível<sup>3</sup>.

Assim, como resultado de parte<sup>4</sup> dos investimentos do *Centro de Pesquisas Avançadas Wernher von Braun*, esta dissertação tem como objetivo determinar o potencial da Computação Quântica Adiabática em abordar o problema de otimização apresentado. Nesse sentido, embora o atual estado da arte dos dispositivos que operam no modelo de CQA não permita a implementação de um algoritmo capaz de otimizar um estoque de médio ou grande porte, nossa proposta é desenvolver um algoritmo quântico para a tarefa e testar sua capacidade para pequenas instâncias. Com isso, será possível avaliar o potencial da CQA na abordagem desse tipo de problema e iniciar uma investigação acerca

<sup>3</sup> Prateleiras de distribuição multinível (*Multi-level rack distribution*) refere-se a um sistema de armazenamento em que os produtos são organizados em vários níveis de prateleiras. Esse tipo de configuração é comumente utilizado para maximizar o uso do espaço vertical disponível, sendo especialmente útil quando se trabalha com produtos que não requerem acesso frequente, pois os itens podem ser armazenados em prateleiras superiores ou inferiores, com os mais frequentemente utilizados posicionados em locais de fácil alcance.

<sup>4</sup> Vale dizer que outros investimentos focam no desenvolvimento de algoritmos clássicos para o problema, também liderados pelo *Centro de Pesquisas Avançadas Wernher von Braun*.

das possíveis vantagens que se pode obter em relação aos algoritmos clássicos.

Portanto, para atender ao objetivo de desenvolver um algoritmo quântico em CQA para o Problema de Gerenciamento de Estoque voltado para fábricas que utilizam prateleiras gravitacionais como sistema padrão de armazenamento, descreveremos no próximo capítulo a estratégia que elaboramos para a tarefa. Essa estratégia envolve dividir a tarefa em duas etapas sequenciais, que serão mapeadas em versões adaptadas de problemas de otimização já descritos na literatura.

A primeira dessas etapas, denominada Fase de Agrupamento, pode ser mapeada em uma versão adaptada do Problema das Múltiplas Mochilas Quadráticas Necessárias, uma versão mais complexa do Problema da Mochila. Já a segunda etapa, denominada Fase de Inserção, pode ser mapeada em uma versão adaptada do Problema do Caixeiro Viajante, previamente apresentado na Seção 3.1.

Com base no princípio de que agrupar insumos frequentemente requisitados juntos nas mesmas prateleiras reduz operações desnecessárias no depósito, ambas as fases da nossa estratégia serão fundamentadas no padrão de ocorrência conjunta dos insumos para determinar quando um par de paletes deve (ou não) ser alocado na mesma prateleira. Para isso, utilizaremos técnicas de *machine learning* para gerar um conjunto de parâmetros que quantificam a ocorrência conjunta entre o  $\alpha$ -ésimo e o  $\beta$ -ésimo paletes a serem inseridos no estoque,  $\lambda_{\alpha\beta}$ . Esses parâmetros serão referidos como parâmetros de correspondência.

Assim, como forma de preparar terreno para apresentarmos nossa estratégia e o algoritmo quântico desenvolvido a partir dela, vamos introduzir na próxima seção o Problema das Múltiplas Mochilas Quadráticas Necessárias, um problema de otimização cuja aplicação mais conhecida é relacionada à busca do melhor arranjo de convidados em mesas durante eventos, e que será necessário para modelarmos a Fase de Agrupamento da estratégia.

## 5.2 Problema das Múltiplas Mochilas Quadráticas

Nessa seção, nosso objetivo é criar bases para formular a estratégia em duas fases desenvolvida para o Problema de Gerenciamento de Estoque<sup>5</sup>. Para isso, nosso primeiro passo é apresentar uma versão mais complexa do Problema da Mochila, o Problema das Múltiplas Mochilas Quadráticas (do inglês, *Multiple Quadratic Knapsack Problem - MQKP*), a partir do qual apresentaremos uma versão levemente modificada, que servirá de base para uma das fases da estratégia.

Como introduzido anteriormente, o Problema da Mochila pode ser enunciado como segue: “Dada uma lista de  $N$  itens únicos, cada um possuindo um determinado valor e

<sup>5</sup> Para evitar repetições, a partir daqui, quando nos referirmos ao Problema de Gerenciamento de Estoque, estamos nos referindo à tarefa adaptada para fábricas que utilizam prateleiras gravitacionais.

peso, quais itens devem ser colocados em uma mochila tal que o peso total seja menor ou igual a uma capacidade máxima  $L$  e o valor carregado seja maximizado?”. Versões mais complexas da tarefa adicionam mais elementos e restrições para que seja possível modelar problemas do mundo real.

No contexto do MQKP [45], em particular, além de serem consideradas múltiplas mochilas, cada uma com sua própria capacidade máxima, também é levado em conta que cada par de itens atribuídos à mesma mochila gera um valor adicional, o qual depende da correspondência entre eles. Dessa forma, o objetivo da tarefa é determinar quais itens devem ser colocados nas mesmas mochilas, respeitando suas capacidades máximas, a fim de maximizar o valor total, que é definido como a soma do valor individual de cada item atribuído a uma mochila e dos valores adicionais gerados pelos pares de itens atribuídos às mesmas mochilas.

Matematicamente, o Problema das Múltiplas Mochilas Quadráticas pode ser posto como segue: “Considere  $M$  mochilas com capacidade máxima de peso  $L_m$  ( $m = 1, \dots, M$ ). Considere uma lista de  $N$  itens únicos, cada qual com seu valor  $v_\alpha$  e peso  $w_\alpha$  ( $\alpha = 1, \dots, N$ ). Considere também que se o  $\alpha$ -ésimo e o  $\beta$ -ésimo itens são atribuídos a mesma mochila, um valor adicional  $\lambda_{\alpha\beta}$  é gerado pela combinação. Dessa forma, se  $x_\alpha^m$  é uma variável binária que representa se o  $\alpha$ -ésimo item está (1) ou não (0) contido na  $m$ -ésima mochila, então o peso e o valor total da  $m$ -ésima mochila são dados por

$$W_m = \sum_{\alpha=1}^N w_\alpha x_\alpha^m, \quad V_m = \sum_{\alpha=1}^N v_\alpha x_\alpha^m + \sum_{\alpha, \beta=1}^N \lambda_{\alpha\beta} x_\alpha^m x_\beta^m, \quad (5.1)$$

respectivamente. A tarefa consiste em maximizar o valor total das mochilas,

$$V = \sum_{m=1}^M V_m, \quad (5.2)$$

respeitando às  $M$  restrições,  $W_m \leq L_m$ ”.

O QMKP pode ser usado para modelar uma ampla variedade de aplicações. Por exemplo, no gerenciamento de projetos, um gerente busca atribuir funcionários a equipes de projeto, e cada projeto tem um orçamento para salários. A produtividade de um projeto depende das capacidades em pares das pessoas atribuídas à equipe, além da produtividade de cada indivíduo. A alocação de funcionários para um conjunto de projetos pode, portanto, ser modelada como um QMKP. Outras aplicações relacionadas incluem a distribuição de recursos em cidades inteligentes [46].

Devido a sua importância prática e relação com outros problemas clássicos de otimização, o QMKP tem recebido cada vez mais atenção da comunidade de pesquisa operacional desde que foi introduzido em 2006 [12]. Na referência, os autores introduziram o problema na literatura e descreveram três algoritmos heurísticos para resolvê-lo, incluindo um algoritmo genético, que foi aprimorado em [92]. Posteriormente, estratégias mais avançadas foram desenvolvidas para a tarefa [93, 94].

### 5.2.1 Problema das Múltiplas Mochilas Quadráticas Necessárias

Ainda que o QMKP já possua similaridades que podem ser exploradas para abordar o Problema de Gerenciamento de Estoque, uma versão modificada e pouco discutida na literatura é de maior interesse para nossos objetivos. Nesta versão, conhecida como Problema das Múltiplas Mochilas Quadráticas Necessárias (do inglês, *Required Multiple Quadratic Knapsack Problem* - RMQKP) [95], todos os itens devem ser alocados nas mochilas. Essa característica, notamos, é muito semelhante à condição de que obrigatoriamente todos os paletes de um depósito devem ser alocados nas prateleiras, isto é, não existem excedentes, como normalmente existem em outras versões do Problema da Mochila.

A formulação matemática do RMQKP não se altera em comparação com a formulação apresentada para o MQKP, eqs. (5.1) e (5.2), além de que a soma do valor dos itens individuais é constante (uma vez que todos os itens devem ser colocados em alguma mochila) e pode, portanto, ser omitida de  $V_m$  tal que

$$V_m = \sum_{\substack{\alpha, \beta=1 \\ \alpha > \beta}}^N \lambda_{\alpha\beta} x_{\alpha}^m x_{\beta}^m. \quad (5.3)$$

Na breve literatura em que é abordado, o RMQKP é conhecido como o Problema do Arranjo de Mesa em Eventos (em inglês, *Table Event Seating Problem* - TESP) [95], estando associado à tarefa de encontrar a melhor forma de distribuir um grupo de  $N$  pessoas em  $M$  mesas, cada uma com  $L_m$  lugares, onde é considerado que o organizador do evento tem uma medida de compatibilidade *ad hoc* para cada par de convidados  $\{\alpha\beta\}$ , que é representada pelo parâmetro de correspondência  $\lambda_{\alpha\beta}$ .

Poucas estratégias já foram desenvolvidas especificamente para abordar o RMQKP (ou, ainda, o TESP). Em específico, na referência [95], encontramos um algoritmo exato [96] desenvolvido para a tarefa. No entanto, o desempenho desse algoritmo cresce exponencialmente com o número de convidados do evento, o que limita sua aplicabilidade a instâncias menores.

Por fim, em acordância com nossos objetivos, é notável que o RMQKP possui características que se assemelham ao Problema de Gerenciamento de Estoque. Assim, com base no conteúdo apresentado até este ponto da dissertação, formularemos no próximo capítulo a tarefa como um problema de otimização em duas fases, sendo uma delas mapeada em uma versão adaptada do RMQKP. Para isso, utilizaremos técnicas de *machine learning* para obter o parâmetro de correspondência  $\lambda_{\alpha\beta}$  entre cada par de paletes a partir dos dados analisados do padrão de ocorrência dos insumos nas requisições da linha de produção. Por último, desenvolveremos para cada fase um hamiltoniano do tipo QUBO que codificará o espaço de soluções da tarefa, a partir do qual poderemos implementar a heurística de *Quantum Annealing* para testar o algoritmo para pequenas instâncias do Problema de Gerenciamento de Estoque.



## 6 Algoritmo Quântico para o Problema de Gerenciamento de Estoque

Nesse capítulo, nosso objetivo é apresentar o algoritmo quântico desenvolvido em Computação Quântica Adiabática para o Problema de Gerenciamento de Estoque conforme especificado até aqui, isto é, adaptado para fábricas que utilizam prateleiras gravitacionais como sistema padrão de armazenamento.

Para isso, nas próximas seções, apresentaremos a estratégia em duas fases elaborada para a tarefa, discutindo como cada uma das fases pode ser mapeada em adaptações de problemas de otimização que foram introduzidos nos capítulos anteriores. Em sequência, para cada uma das fases, construiremos um Hamiltoniano Problema, possibilitando a implementação da estratégia nos *quantum annealers* da *D-Wave*. Finalmente, para atestar o funcionamento do algoritmo, que consiste na aplicação sequencial das fases, implementaremos a estratégia para um conjunto de instâncias simples nestes dispositivos.

### 6.1 Adaptando o RMQKP para o Gerenciamento de Estoque

Antes de avançarmos para o desenvolvimento da estratégia em duas fases elaborada para o Problema de Gerenciamento de Estoque, apresentamos nessa seção uma versão que adaptamos do Problema das Múltiplas Mochilas Quadráticas Necessárias (RMQKP) para o contexto de gestão de depósito. Mais adiante, essa adaptação será utilizada para mapear a Fase de Agrupamento da nossa estratégia.

Com esse propósito, vamos modelar a tarefa de distribuir um determinado número de paletes em um determinado número de prateleiras (por enquanto, sem considerar a ordem de inserção), onde a alocação conjunta de cada par de paletes gera um valor adicional. Adicionalmente, introduzimos um novo elemento: Em cada uma das prateleiras, existe um conjunto de paletes já alocados, de tal forma que ao alocar novos paletes também são gerados valores adicionais da interação entre os novos paletes alocados e os paletes previamente posicionados<sup>1</sup>.

Os valores adicionais gerados são parâmetros de correspondência que quantificam quão recorrentemente o conteúdo entre dois paletes é solicitado em conjunto pela linha de produção. O objetivo da tarefa que modelaremos a seguir é distribuir os paletes entre as prateleiras de maneira a agrupar aqueles que mais recorrentemente são solicitados

---

<sup>1</sup> Nesse sentido, definimos uma variação do Problema da Mochila, a qual não encontramos registros na literatura analisada.

em conjunto, maximizando assim a eficiência da alocação, enquanto são respeitadas as restrições de capacidade das prateleiras.

Para modelar a tarefa, consideramos  $N$  como o número de novos paletes que desejamos alocar nas prateleiras gravitacionais de um depósito, e  $P$  como o número de paletes já distribuídos entre essas prateleiras. A cada um dos  $N$  novos paletes, associamos um custo-prateleira  $c_\alpha$  ( $\alpha = 1, \dots, N$ ), que representa o custo atribuído à prateleira em que o  $\alpha$ -ésimo paleta será alocado, devido a sua presença. Esse custo pode representar, por exemplo, o peso ou o comprimento do paleta.

Consideramos  $M$  como o número de prateleiras gravitacionais disponíveis no depósito. A cada prateleira, associamos um limite  $L_m$  ( $m = 1, \dots, M$ ), que pode representar, por exemplo, a capacidade máxima de peso suportado ou a profundidade máxima para a alocação de paletes da  $m$ -ésima prateleira. Similarmente, definimos  $R_m$  como o limite ainda disponível para alocar os  $N$  novos paletes na  $m$ -ésima prateleira, ou seja, como a diferença entre  $L_m$  e a soma dos custos-prateleiras dos  $P_m$  paletes já ali alocados.

O significado físico de  $c_\alpha$  deve ser coerente com  $R_m$ . Por exemplo, se  $R_m$  denota o limite de peso ainda suportado pela  $m$ -ésima prateleira, então  $c_\alpha$  deve representar o peso do  $\alpha$ -ésimo paleta. Se  $R_m$  denota a profundidade ainda disponível na  $m$ -ésima prateleira, então  $c_\alpha$  deve representar o comprimento do  $\alpha$ -ésimo paleta. Ainda, em um exemplo de particular interesse, se  $R_m$  denota a quantidade de paletes que ainda podem ser alocados na  $m$ -ésima prateleira, tem-se que  $c_\alpha = 1$ ,  $\forall \alpha$ , de modo que o custo-prateleira de cada novo paleta seja igual a uma unidade de espaço ocupado.

Matematicamente, as variáveis estão relacionadas pela restrição de que, ao alocar novos paletes na  $m$ -ésima prateleira, a soma dos custos  $c_\alpha$  não deve exceder o limite  $R_m$ , isto é, a desigualdade

$$\sum_{\alpha=1}^N c_\alpha x_\alpha^m \leq R_m, \quad (6.1)$$

onde  $x_\alpha^m$  é uma variável binária que indica se o  $\alpha$ -ésimo paleta está (1) ou não (0) alocado na  $m$ -ésima prateleira gravitacional.

Além disso, para cada possível par de paletes  $\{\alpha\beta\}$  a serem alocados no depósito, atribuímos um parâmetro de correspondência  $\lambda_{\alpha\beta}$ ,  $0 \leq \lambda_{\alpha\beta} \leq 1$ , que quantifica quão recorrente é a requisição conjunta dos conteúdos desses paletes nas demandas. Definimos esse parâmetro da seguinte forma: quão maior a prioridade em alocar o par de paletes  $\{\alpha\beta\}$  na mesma prateleira, menor o valor de  $\lambda_{\alpha\beta}$ .

Notemos que, nesse sentido, o parâmetro definido é visto como um custo associado ao depósito quando dois paletes são alocados na mesma prateleira. Em outras palavras, quão menor a soma de todos os custos  $\{\alpha\beta\}$  dos paletes alocados nas mesmas prateleiras, mais eficiente é a organização do depósito. Por essa razão, daqui em diante, vamos nos referir ao parâmetro de correspondência  $\lambda_{\alpha\beta}$  como o custo-interpaleta entre o  $\alpha$ -ésimo e o

$\beta$ -ésimo palete.

A abordagem assumida indica que, para  $\alpha \neq \beta$ , se (i)  $\lambda_{\alpha\beta} = 0$ , então o par de paletes deve preferencialmente ser alocado na mesma prateleira e, se (ii)  $\lambda_{\alpha\beta} = 1$ , então o par de paletes deve preferencialmente ser alocado em prateleiras diferentes. Valores de  $\lambda_{\alpha\beta}$  entre esses extremos indicam quão prioritário é que estes paletes sejam alocados nas mesmas prateleiras em detrimento de outras combinações. Para  $\alpha = \beta$ , definimos<sup>2</sup>  $\lambda_{\alpha\beta} = 0$ .

De forma análoga aos custos-interpaletes  $\lambda_{\alpha\beta}$  entre os  $N$  novos paletes que devem ser distribuídos entre as prateleiras do depósito, definimos também os custos-interpaletes  $\lambda_{\alpha\tau}^{(m)}$ , com  $(\tau = 1, \dots, P_m)$ , entre os  $N$  paletes a serem alocados e os  $P_m$  paletes previamente alocados na  $m$ -ésima prateleira. Dessa maneira, ao alocar o  $\alpha$ -ésimo paleta na  $m$ -ésima prateleira, são gerados  $P_m$  custos-interpaletes, cuja soma é igual a  $\sum_{\tau=1}^{P_m} \lambda_{\alpha\tau}^{(m)}$ .

Assim, com as variáveis definidas, a tarefa de distribuir um conjunto de novos paletes nas prateleiras do depósito (sem se preocupar com a ordem de inserção, por enquanto) pode ser formulada como segue: Considere  $N$  paletes a serem inseridos no depósito, a cada qual associado um custo-prateleira  $c_\alpha$ , com  $(\alpha = 1, \dots, N)$ . Considere  $M$  prateleiras gravitacionais, cada qual com  $P_m$  paletes previamente alocados, e um limite  $R_m$  ainda disponível para alocação de novos paletes, com  $(m = 1, \dots, M)$ . Considere também que um custo-interpaleta  $\lambda_{\alpha\beta}$  é gerado se o  $\alpha$ -ésimo e o  $\beta$ -ésimo paletes são alocados nas mesmas prateleiras, com  $(\beta = 1, \dots, N)$ , e que  $P_m$  custos-interpaletes  $\lambda_{\alpha\tau}^{(m)}$  são gerados se o  $\alpha$ -ésimo paleta é alocado na  $m$ -ésima prateleira, com  $(\tau = 1, \dots, P_m)$ . Dessa forma, se  $x_\alpha^m$  é uma variável binária que indica se o  $\alpha$ -ésimo paleta está (1) ou não (0) alocado na  $m$ -ésima prateleira, então a soma dos custos-prateleiras e a soma dos custos-interpaletes na  $m$ -ésima prateleira são respectivamente dados por

$$C_m = \sum_{\alpha=1}^N c_\alpha x_\alpha^m, \quad \Lambda_m = \sum_{\substack{\alpha, \beta=1 \\ \alpha > \beta}}^N \lambda_{\alpha\beta} x_\alpha^m x_\beta^m + \sum_{\alpha=1}^N \sum_{\tau=1}^{P_m} \lambda_{\alpha\tau}^{(m)} x_\alpha^m. \quad (6.2)$$

O objetivo da tarefa consiste em minimizar o custo-interpaleta total

$$\Lambda = \sum_{m=1}^M \Lambda_m, \quad (6.3)$$

de modo a agrupar nas mesmas prateleiras os paletes que mais recorrentemente são solicitados em conjunto nas demandas da linha de produção, enquanto é assegurado que todos os  $N$  novos paletes são alocados nas prateleiras do depósito e que as  $M$  restrições de capacidade  $C_m \leq R_m$  são respeitadas.

<sup>2</sup> Se  $\alpha = \beta$ , estamos nos referindo ao mesmo paleta duas vezes. Não faz sentido, portanto, definir um custo que não seja nulo.

## 6.2 Obtenção dos Parâmetros via *Machine Learning*

A seguir, apresentamos de forma concisa a abordagem de *machine learning* empregada para gerar os custos-interpaletes  $\lambda_{\alpha\beta}$ , presentes na eq. (6.2), que quantificam quão recorrente é a requisição conjunta do conteúdo do  $\alpha$ -ésimo e  $\beta$ -ésimo paletes nas demandas da linha de produção. Os resultados descritos a seguir são de autoria da aluna de mestrado do PPGF, Amanda Valério, que também integra o projeto desenvolvido em parceria com o *Centro de Pesquisas Avançadas Wernher von Braun*, sendo responsável pela análise dos dados da fábrica parceira.

Para gerar os custos-interpaletes, foram utilizadas técnicas de *machine learning* para analisar os registros das demandas da linha de produção sobre todos os 144 tipos de insumos utilizados pela fábrica, por um total de 33 dias. Especificamente, foi empregada uma modificação do Algoritmo de Recomendação *Apriori* [97], uma ferramenta utilizada para gerar recomendações ao identificar padrões presentes em um conjunto de dados, por meio de regras de associação.

Regras de associação são parâmetros numéricos que identificam uma relação entre diferentes itens em um conjunto de dados, gerando recomendações [98]. Nesse contexto, utilizamos a métrica de Confiança, que representa a proporção entre o número de vezes em que o  $a$ -ésimo e  $b$ -ésimo insumos foram requisitados na mesma demanda ( $a \cap b$ ) e o número total de vezes que  $a$ -ésimo insumo foi requisitado, isto é,

$$\text{Confiança}(a, b) = \frac{\text{número de requisições contendo } a \text{ e } b}{\text{número de requisições contendo } a}. \quad (6.4)$$

Nesse sentido, a Confiança mede a proporção de vezes que a recomendação  $a$  implica  $b$  é verdadeira nos registros analisados. Ao aplicar o Algoritmo Apriori, os valores de Confiança foram gerados para todos os possíveis pares de insumos utilizados na fábrica, e armazenados em uma matriz.

A partir da matriz de Confiança, é possível determinar o custo-interpalete entre quaisquer paletes no depósito. Para isso, assumindo que o  $\alpha$ -ésimo ( $\beta$ -ésimo) palete contém o  $a$ -ésimo ( $b$ -ésimo) insumo, temos que o custo-interpalete entre o par  $\{\alpha\beta\}$  é dado por

$$\lambda_{\alpha\beta} = 1 - \frac{\text{Confiança}(a, b) + \text{Confiança}(b, a)}{2}, \quad (6.5)$$

onde a média entre os valores de Confiança é realizada para que uma matriz contendo todos os custos-interpaletes seja simétrica. Uma unidade é subtraída para que  $\lambda_{\alpha\beta}$  seja mínimo (máximo) quando a correspondência entre os paletes for máxima (mínima), como anteriormente definido. Os custos-interpaletes entre os  $N$  novos paletes a serem distribuídos nas prateleiras e os  $P_m$  paletes previamente alocados em cada prateleira,  $\lambda_{\alpha\tau}^{(m)}$ , são gerados de forma análoga.

## 6.3 Estratégia em Duas Fases para o Problema de Gerenciamento de Estoque

Nessa seção, apresentamos de forma detalhada a estratégia elaborada para abordar o Problema de Gerenciamento de Estoque como delineado até aqui. Nosso objetivo é alocar  $N$  novos paletes nas  $M$  prateleiras gravitacionais disponíveis no depósito, cada uma com  $P_m$  paletes já previamente alocados, de tal forma que o número de reinserções necessárias nas futuras operações do depósito seja minimizado.

Com esse propósito, nossa estratégia é dividida em duas etapas distintas. Na primeira etapa, os  $N$  paletes são separados em  $M$  grupos, cada grupo destinado a uma prateleira gravitacional específica. Na segunda etapa, ordem de inserção dos paletes de cada grupo em suas respectivas prateleiras é determinada, seguindo o princípio de gerenciamento FIFO. Assim, a estratégia aqui desenvolvida divide a tarefa em duas fases sequenciais, que nomeamos como Fase de Agrupamento e Fase de Inserção, as quais detalhamos a seguir.

Na Fase de Agrupamento, temos como objetivo distribuir os  $N$  novos paletes em  $M$  grupos de tal forma que o  $m$ -ésimo grupo corresponda ao conjunto de paletes que será alocado na  $m$ -ésima prateleira, respeitando as  $M$  restrições de capacidade expressas na eq. (6.1). Para tanto, são utilizados os parâmetros gerados a partir dos registros das demandas da linha de produção a fim de minimizar o número de reinserções necessárias nas futuras operações do depósito de tal forma que a função objetivo dada por

$$\Lambda = \sum_{m=1}^M \left( \sum_{\substack{\alpha, \beta=1 \\ \alpha > \beta}}^N \lambda_{\alpha\beta} x_{\alpha}^m x_{\beta}^m + \sum_{\alpha=1}^N \sum_{\tau=1}^{P_m} \lambda_{\alpha\tau}^{(m)} x_{\alpha}^m \right) \quad (6.6)$$

seja minimizada, como discutido na Seção 6.1. Os custos-interpaletes  $\lambda_{\alpha\beta}$  e  $\lambda_{\alpha\tau}^{(m)}$  são gerados a partir de técnicas de *machine learning*, como discutido na Seção 6.2.

Minimizando a eq. (6.6), temos como resultado  $M$  conjuntos de paletes, agrupados de tal modo que os paletes com maior correspondência entre si (e levando em conta também aqueles previamente alocados) são destinados às mesmas prateleiras. Notemos, ainda, que seguindo o princípio de gerenciamento FIFO, no qual os paletes são retirados na mesma ordem em que são inseridos, é também necessário determinar a ordem de inserção de cada conjunto em sua prateleira correspondente.

Portanto, na etapa seguinte, que chamamos de Fase de Inserção, temos como objetivo determinar a ordem em que os paletes do  $m$ -ésimo grupo devem ser inseridos na  $m$ -ésima prateleira gravitacional. Para isso, adotamos a abordagem de posicionar os paletes em uma sequência que minimiza a soma dos custos-interpaletes entre os vizinhos adjacentes, em uma estratégia inspirada no Problema do Caixeiro Viajante, de tal forma que a organização interna de cada prateleira seja aquela em que (de forma global) os

insumos requisitados conjuntamente com maior recorrência estejam o mais próximo possível um dos outros.

Assim, para determinar a sequência de inserção do  $m$ -ésimo conjunto de paletes na  $m$ -ésima prateleira, onde já estão previamente alocados  $P_m$  paletes, a função objetivo a ser minimizada é dada por

$$\Lambda'_m = \sum_{\substack{\alpha, \beta=1 \\ \alpha > \beta}}^{N_m} \sum_{s=P_m+2}^{P_m+N_m} \lambda_{\alpha\beta} x_\alpha^{(s)} x_\beta^{(s-1)} + \sum_{\alpha=1}^{N_m} \lambda_{\alpha\gamma} x_\alpha^{(P_m+1)}, \quad (6.7)$$

onde  $N_m$  é o número de paletes a serem inseridos e  $s$  é o índice introduzido para enumerar as posições dos paletes nas prateleiras<sup>3</sup>. O parâmetro  $\lambda_{\alpha\gamma}$  denota o custo-interpalete entre o palete destinado à posição  $s = P_m + 1$  e o último dos  $P_m$  paletes, que ocupa a posição  $s = P_m$ .

Na eq. (6.7), o primeiro termo determina a sequência de inserção que minimiza a soma dos custos-interpaletes adjacentes entre os paletes nas posições entre  $s = P_m + 2$  e  $s = P_m + N_m$  enquanto o segundo termo determina o palete que deve ocupar a posição  $s = P_m + 1$ , estando adjacente ao palete da posição  $s = P_m$ , o último dos  $P_m$  paletes já previamente alocados.

Nesse sentido, definido qual o primeiro palete a ser inserido, nossa abordagem para a Fase de Inserção difere da estratégia mais intuitiva de inserir como segundo palete aquele com maior correspondência com o primeiro, como terceiro aquele com maior correspondência com o segundo, e assim por diante. Aqui, ao utilizar a minimização da soma total dos custos interpaletes entre os vizinhos adjacentes como critério de otimização, a ordenação é otimizada considerando todos os itens do grupo simultaneamente, garantindo uma estratégia mais confiável para a ordenação interna dos paletes dentro das prateleiras.

## 6.4 Algoritmo Quântico para o Problema de Gerenciamento de Estoque

Com base na estratégia elaborada na seção anterior, apresentaremos a seguir o algoritmo quântico desenvolvido para abordar o Problema de Gerenciamento de Estoque. Para isso, construiremos um Hamiltoniano Problema para cada uma das fases da estratégia, possibilitando sua implementação em Computação Quântica Adiabática.

<sup>3</sup> Dessa maneira,  $s = 1$  denota a posição já ocupada pelo primeiro dos  $P_m$  paletes previamente alocados,  $s = P_m + 1$  denota a posição a ser ocupada pelo primeiro dos  $N_m$  novos paletes e  $s = P_m + N_m$  denota a última posição a ser ocupada pelo último dos  $N_m$  paletes.

## Hamiltoniano para o Problema de Gerenciamento de Estoque

### Hamiltoniano para a Fase de Agrupamento

Para a Fase de Agrupamento, vamos propor um hamiltoniano da forma

$$H_{agrup} = H_A + H_B + H_C, \quad (6.8)$$

onde cada autoestado codifica uma possível maneira de distribuir os  $N$  novos paletes entre as  $M$  prateleiras gravitacionais<sup>4</sup> do depósito, nas quais já estão distribuídos  $P$  paletes previamente alocados. Para isso, com a ideia de utilizar o Teorema Adiabático para encontrar soluções de alta qualidade, vamos codificar os autoestados de tal forma que quão maior a qualidade da solução codificada, menor a autoenergia a ela associada.

Em primeiro lugar, queremos construir um hamiltoniano que assegure que cada um dos  $N$  novos paletes seja alocado em uma (e apenas uma) prateleira. Para isso, vamos introduzir a variável binária  $x_\alpha^m$ , onde  $(\alpha = 1, \dots, N)$  e  $(m = 1, \dots, M)$ , tal que

$$x_\alpha^m = \begin{cases} 1, & \text{caso o } \alpha\text{-ésimo palete seja alocado na } m\text{-ésima prateleira,} \\ 0, & \text{caso contrário,} \end{cases} \quad (6.9)$$

e consideremos o hamiltoniano

$$H_A = A \sum_{\alpha=1}^N \left( 1 - \sum_{m=1}^M x_\alpha^m \right)^2, \quad (6.10)$$

que introduz uma penalização energética proporcional a  $A$  às soluções em que o  $\alpha$ -ésimo palete está alocado em mais que uma (ou nenhuma) prateleira, distanciando essas soluções do mínimo global.

Em segundo lugar, para realizar a tarefa de distribuir eficientemente os  $N$  novos paletes entre as  $M$  prateleiras gravitacionais, com  $P_m$  paletes já previamente alocados em cada uma, de forma a minimizar o custo interpalete total  $\Lambda$ , consideremos um hamiltoniano da forma

$$H_B = B \sum_{m=1}^M \left( \sum_{\substack{\alpha, \beta=1 \\ (\alpha > \beta)}}^N \lambda_{\alpha\beta} x_\alpha^m x_\beta^m + \sum_{\alpha=1}^N \sum_{\tau=1}^{P_m} \lambda_{\alpha\tau}^{(m)} x_\alpha^m \right), \quad (6.11)$$

que codifica em seu espectro a função objetivo introduzida na eq. (6.6), onde os custos-interpaletes  $\lambda_{\alpha\beta}$  e  $\lambda_{\alpha\tau}^{(m)}$  são gerados via *machine learning* como descrito na Seção 6.2. O parâmetro  $B$  deve ser ajustado em relação aos parâmetros de penalização dos demais termos de  $H_{agrup}$ .

Vale notar que o estado fundamental de  $H_B$  é aquele em que nenhum palete é destinado a uma prateleira, isto é,  $x_\alpha^m = 0, \forall \alpha, m$ . Ainda, estados próximos ao estado

<sup>4</sup> Preferimos aqui a exposição direta da fase como a tarefa de distribuir  $N$  paletes em  $M$  prateleiras gravitacionais em vez de  $M$  grupos, embora os termos sejam intercambiáveis e, portanto, a ideia seja idêntica.

fundamental codificam soluções em que nem todos os paletes são alocados. De fato, o que impede que configurações como essas sejam soluções de baixa energia também no hamiltoniano  $H_{agrup}$  é a presença do termo  $H_A$ , que penaliza soluções em que um palete (ou mais) não são destinados a uma prateleira, reforçando assim sua importância.

Por fim, o último hamiltoniano que introduzimos é dado por

$$H_C = C \sum_{m=1}^M \left( \sum_{\alpha=1}^N c_\alpha x_\alpha^m + \langle \mathbf{2} | \mathbf{a}_m \rangle - R_m \right)^2, \quad (6.12)$$

onde utilizamos a notação

$$\langle \mathbf{2} | \mathbf{a}_m \rangle = \sum_{l=0}^{\Omega_m} 2^l a_l^m, \quad (6.13)$$

com  $\Omega_m = \lfloor \log_2 R_m \rfloor$ , para representar a expansão binária sobre os qubits auxiliares  $a_l^m$ , que são definidos de forma idêntica à  $x_l^m$ , mas representam paletes virtuais com custo  $c_\alpha = 2^l$ , preenchendo a  $m$ -ésima prateleira apenas virtualmente.

Nesse sentido, a expansão binária em termos dos qubits auxiliares tem como objetivo completar a diferença entre o somatório dos custos-prateleiras dos paletes alocados e o limite  $R_m$  da prateleira, permitindo que a desigualdade introduzida na eq. (6.1) possa ser satisfeita. Notemos que a ausência dessa expansão faria com que existisse uma tendência de que paletes reais de custo  $c_\alpha$  fossem alocados nas mesmas prateleiras até que o limite  $R_m$  fosse atingido. Em outras palavras, a expansão binária permite que existam soluções nas quais a soma dos custos-prateleiras dos itens não seja necessariamente igual ao limite da prateleira.

Logo, o hamiltoniano final é dado por

$$H_{agrup} = A \sum_{\alpha=1}^N \left( 1 - \sum_{m=1}^M x_\alpha^m \right)^2 + B \sum_{m=1}^M \left( \sum_{\substack{\alpha, \beta=1 \\ (\alpha > \beta)}}^N \lambda_{\alpha\beta} x_\alpha^m x_\beta^m + \sum_{\alpha=1}^N \sum_{\tau=1}^{P_m} \lambda_{\alpha\tau}^{(m)} x_\alpha^m \right) + C \sum_{m=1}^M \left( \sum_{\alpha=1}^N c_\alpha x_\alpha^m + \langle \mathbf{2} | \mathbf{a}_m \rangle - R_m \right)^2, \quad (6.14)$$

com autoestados da forma  $|x_1^1, \dots, x_1^M, \dots, x_N^1, \dots, x_N^M, a_0^1, \dots, a_0^M, a_{\Omega_m}^1, \dots, a_{\Omega_m}^M\rangle$ , sendo que o estado fundamental corresponde a melhor solução da tarefa. Vale ressaltar que  $H_{agrup}$  é também um hamiltoniano do tipo QUBO e, portanto, pode ser implementado nos *quantum annealers* da *D-Wave*, sendo necessários  $N(M + \lfloor \log_2 R_m \rfloor + 1)$  qubits lógicos para implementá-lo. Notemos que o número de qubits lógicos não depende do número de paletes pré-alocados em cada prateleira,  $P_m$ .

### Hamiltoniano para a Fase de Inserção

Como resultado da implementação da Fase de Agrupamento, devemos obter um estado fundamental que codifica a forma mais eficiente de distribuir  $N$  paletes em  $M$

grupos, sendo que o  $m$ -ésimo grupo corresponde ao conjunto de paletes destinado a ser alocado na  $m$ -ésima prateleira gravitacional. Os primeiros estados excitados, de forma similar, codificam soluções de alta qualidade.

Na Fase de Inserção, nosso objetivo é determinar a sequência com que os paletes de cada conjunto devem ser inseridos nas prateleiras a quais foram destinados. Sendo assim, como a tarefa de determinar a sequência de inserção do  $m$ -ésimo conjunto de paletes na  $m$ -ésima prateleira independe das demais  $(M - 1)$  configurações de paletes e prateleiras, dividimos a fase em  $M$  subtarefas, que consistem em aplicar o hamiltoniano que desenvolveremos para cada um dos  $M$  conjuntos obtidos na Fase de Agrupamento.

Para que isso seja possível, definimos como  $N_m$  o número de paletes no  $m$ -ésimo grupo e introduzimos a variável binária  $x_{\alpha,s}$ , com  $(\alpha = 1, \dots, N_m)$  e  $(s = P_m + 1, \dots, P_m + N_m)$ , tal que (para a  $m$ -ésima prateleira)

$$x_{\alpha}^{(s)} = \begin{cases} 1, & \text{caso o } \alpha\text{-ésimo palete seja inserido na } s\text{-ésima posição,} \\ 0, & \text{caso contrário,} \end{cases} \quad (6.15)$$

e consideramos o hamiltoniano

$$H_{ins} = \sum_{\substack{\alpha, \beta=1 \\ \alpha > \beta}}^{N_m} \sum_{s=P_m+2}^{P_m+N_m} \lambda_{\alpha\beta} x_{\alpha}^{(s)} x_{\beta}^{(s-1)} + \sum_{\alpha=1}^{N_m} \lambda_{\alpha\gamma} x_{\alpha}^{(P_m+1)} + D \left[ \sum_{s=P_m+1}^{P_m+N_m} \left( 1 - \sum_{\alpha=1}^{N_m} x_{\alpha}^{(s)} \right) + \sum_{\alpha=1}^{N_m} \left( 1 - \sum_{s=P_m+1}^{P_m+N_m} x_{\alpha}^{(s)} \right) \right], \quad (6.16)$$

onde os dois primeiros termos codificam no espectro a função objetivo introduzida na eq. (6.7) e o último termo adiciona uma penalização energética proporcional a  $D$  às soluções em que um palete é alocado em mais que uma (ou nenhuma) posição da  $m$ -ésima prateleira, de forma similar ao que é feito na aplicação do Problema do Caixeiro Viajante<sup>5</sup>.

Ainda, com o estado fundamental correspondendo à melhor solução da tarefa, os autoestados de  $H_{ins}$  são da forma  $|x_1^{(P_m+1)}, \dots, x_{N_m}^{(P_m+1)}, \dots, x_1^{(P_m+N_m)}, \dots, x_{N_m}^{(P_m+N_m)}\rangle$ . Vale também ressaltar que  $H_{ins}$  é um hamiltoniano do tipo QUBO e, portanto, pode ser implementado nos *quantum annealers* da *D-Wave*, sendo necessários  $(N_m)^2$  qubits lógicos. Novamente, notemos que o número de qubits lógicos necessários não depende do número de paletes pré-alocados em cada prateleira,  $P_m$ .

<sup>5</sup> Nesse sentido, existe uma diferença essencial entre  $H_{ins}$  e aquele apresentado para o Problema do Caixeiro Viajante na eq. (4.9). No atual contexto, estamos interessados em determinar qual o menor caminho (ou, em outras palavras, qual a sequência de inserção dos paletes que minimiza a soma dos custos-interpaletes entre os vizinhos adjacentes) em vez de a menor rota. Para essa adaptação, alteramos os índices do somatório em  $s$  no primeiro termo para não incluir conexões entre o primeiro e o último palete a serem inseridos na  $m$ -ésima prateleira, evitando a formação de um ciclo.

## Passo-a-Passo do Algoritmo Quântico Desenvolvido

Em suma, a aplicação do algoritmo quântico desenvolvido para a distribuição eficiente de  $N$  paletes em  $M$  prateleiras gravitacionais deve ser realizada da seguinte forma:

1. Os registros das demandas da linha de produção de uma fábrica são analisados via *machine learning*, gerando os custos-interpaletes, como discutido na Seção 6.2.
2. A Fase de Agrupamento é realizada em CQA, implementando  $H_{agrup}$  para determinar a forma mais eficiente de distribuir  $N$  paletes em  $M$  grupos, de acordo com nossa estratégia. Cada grupo corresponde ao conjunto de paletes a ser inserido na  $m$ -ésima prateleira.
3. Em sequência, a Fase de Inserção é realizada em CQA, implementando  $H_{ins}$  para determinar a forma mais eficiente de inserir os paletes do  $m$ -ésimo grupo na  $m$ -ésima prateleira, de acordo com nossa estratégia.

Por fim, se em ambas as fases são medidos os estados fundamentais de  $H_{agrup}$  e  $H_{ins}$ , nos quais estão codificadas as melhores soluções de cada fase, obtém-se a disposição mais eficiente do depósito, de acordo com nossa estratégia. Da forma com que foram construídos os hamiltonianos, estados na vizinhança do mínimo global codificam também soluções de alta qualidade.

## 6.5 Implementação do Algoritmo Quântico para o Problema de Gerenciamento de Estoque em *Quantum Annealing*

Na última seção, cumprimos o objetivo central dessa dissertação: desenvolver um algoritmo quântico em Computação Quântica Adiabática para o Problema de Gerenciamento de Estoque. A seguir, devido ao tempo limitado do projeto, implementaremos o algoritmo para o cenário mais simples possível: aquele em que não existem paletes previamente alocados nas prateleiras gravitacionais do depósito.

Para um depósito inicialmente vazio, os hamiltonianos expressos pelas eqs. (6.14) e (6.16) se reduzem às versões simplificadas

$$\begin{aligned}
 H_{agrup} = & A \sum_{\alpha=1}^N \left( 1 - \sum_{m=1}^M x_{\alpha}^m \right)^2 + B \sum_{\substack{\alpha, \beta=1 \\ (\alpha > \beta)}}^N \lambda_{\alpha\beta} \sum_{m=1}^M x_{\alpha}^m x_{\beta}^m \\
 & + C \sum_{m=1}^M \left( \sum_{\alpha=1}^N c_{\alpha} x_{\alpha}^m + \langle \mathbf{2} | \mathbf{a}_m \rangle - L_m \right)^2, \tag{6.17}
 \end{aligned}$$

$$H_{ins} = \sum_{\substack{\alpha, \beta=1 \\ (\alpha > \beta)}}^{N_m} \lambda_{\alpha\beta} \sum_{s=2}^{N_m} x_{\alpha}^{(s)} x_{\beta}^{(s-1)} + D \left[ \sum_{s=1}^{N_m} \left( 1 - \sum_{\alpha=1}^{N_m} x_{\alpha}^{(s)} \right) + \sum_{\alpha=1}^{N_m} \left( 1 - \sum_{s=1}^{N_m} x_{\alpha}^{(s)} \right) \right], \quad (6.18)$$

onde  $L_m$  representa o limite da  $m$ -ésima prateleira gravitacional, ao invés de  $R_m$ , uma vez que não há paletes já alocados nas prateleiras.

Em sequência, com as versões simplificadas dos Hamiltonianos Problemas conforme apresentadas, implementaremos o algoritmo quântico desenvolvido para o Problema de Gerenciamento de Estoque para abordar um conjunto de instâncias simples nos *quantum annealers* da *D-Wave*, discutindo aspectos relacionados ao atual estado da arte dos dispositivos, que restringem o tamanho dos exemplos que podem ser tratados.

Nos resultados que mostraremos a seguir, definiremos o limite  $L_m$  como o número máximo de paletes suportados pela  $m$ -ésima prateleira, com  $c_{\alpha} = 1, \forall \alpha$ , de tal forma que o custo-prateleira de cada palete seja igual a uma unidade de espaço ocupada. Escolhemos esse exemplo em particular, pois, entre as possíveis aplicações, essa é a de maior interesse imediato para a fábrica parceira do projeto.

Em particular, em relação a escolha dos parâmetros de penalização para a Fase de Agrupamento, os parâmetros utilizados são dados por  $C = (1 + \max(\lambda_{\alpha\beta}))\sqrt{N}$ ,  $B = C / \max(\lambda_{\alpha\beta})$  e  $A = C^2$ . A escolha desses parâmetros foi feita a partir de testes nos quais comparamos outros valores de parâmetros e determinamos que estes se mostraram mais apropriados, uma vez que permitiram reduzir o número de *shots* necessários para encontrar as melhores soluções para as instâncias analisadas. Mantivemos o tempo de *annealing* igual a 20  $\mu s$  em todos os testes.

Ainda, antes de discorrermos sobre os resultados obtidos, enfatizamos que nessa última seção anterior à apresentação das conclusões desse projeto, nosso objetivo se resume a testar o funcionamento do algoritmo e, portanto, dos hamiltonianos propostos, preparando terreno para futuros trabalhos, onde uma análise mais profunda e completa dos potenciais resultados dessa dissertação serão explorados.

### 6.5.1 Depósito com Duas Prateleiras Gravitacionais, com $L_m = 2$

Com o propósito de verificar o funcionamento do algoritmo, vamos abordar a simples tarefa de alocar um conjunto de 3 a 4 paletes em um depósito com 2 prateleiras gravitacionais, com 2 espaços em cada.

Como primeiro exemplo, consideremos o caso em que  $N = 3$  paletes devem ser eficientemente distribuídos nas prateleiras gravitacionais do depósito. Para essa tarefa, precisamos considerar os  $N^2$  custos-interpaletes  $\lambda_{\alpha\beta}$  envolvidos<sup>6</sup>, que são dados pelos

<sup>6</sup> Por clareza, registramos aqui que os elementos da matriz  $\lambda$  nesses dois primeiros exemplos são artificiais,

elementos da matriz  $3 \times 3$  simétrica

$$\lambda = \begin{pmatrix} 0 & 0.4 & 0.2 \\ 0.4 & 0 & 0.6 \\ 0.2 & 0.6 & 0 \end{pmatrix},$$

a partir da qual se pode perceber que, ao distribuir os paletes entre as prateleiras, espera-se que o primeiro e o terceiro paletes sejam conjuntamente alocados (agrupados) na prateleira A, enquanto o segundo patele é alocado individualmente na prateleira B, uma vez que  $\lambda_{13} < \lambda_{12} < \lambda_{23}$ . Além disso, como as prateleiras são indistinguíveis ( $A \rightleftharpoons B$ ), existem duas soluções ótimas para a configuração.

Dado que já conhecemos as soluções ótimas antecipadamente, vamos em seguida implementar o algoritmo desenvolvido nessa dissertação nos *quantum annealers* da *D-Wave*. Com isso, poderemos comparar as soluções obtidas pelo algoritmo com as soluções ótimas previamente conhecidas, verificando se as melhores soluções são, de fato, encontradas.

Na Fase de Agrupamento, o hamiltoniano expresso pela eq. (6.17) foi implementado no *quantum annealer D-Wave Advantage 4.1* para o caso em que queremos distribuir eficientemente  $N = 3$  paletes, com parâmetros de correspondência dados pela matriz  $\lambda$ , em  $M = 2$  prateleiras com  $L_m = 2$  espaços disponíveis em cada. As soluções esperadas, obtidas da análise da matriz  $\lambda$ , estão codificadas nos estados<sup>7</sup> de *bitstrings* 0110011000 (408, em decimal) e 1001100100 (612, em decimal), dada a dupla degenerescência do estado fundamental do hamiltoniano para codificar ambas as soluções ótimas.

Os resultados obtidos na implementação mostram que, para uma amostra com 100 *shots*, os estados fundamentais do hamiltoniano aparecem em média 35,4 vezes, alcançando assim a melhor solução da Fase de Agrupamento. Foram necessários 12 qubits físicos para mapear os 10 qubits lógicos necessários para implementação da tarefa na topologia do *quantum annealer*. O tratamento estatístico foi feito com 50 *runs*.

Nas Figuras 8 e 9, uma das 50 amostras com 100 *shots* é analisada. Na Figura 8, o espectro do hamiltoniano é representado graficamente, com os pontos em vermelho indicando os dez estados de maior ocorrência na amostra. Nota-se que todos esses pontos representam estados de mínimos locais (ou globais) do espectro. Na Figura 9, a energia e o número de ocorrências desses estados são apresentados, mostrando que os dois estados mais medidos são aqueles que codificam o estado fundamental duplamente degenerado da Fase de Agrupamento, encontrados em 45% dos *shots* da amostra.

---

isto é, não advém das técnicas de *machine learning* e foram usados para testar o algoritmo de forma mais básica. Em seguida desses exemplos, trataremos um conjunto de instâncias onde as matrizes utilizadas são obtidas a partir dessas técnicas.

<sup>7</sup> Os autoestados de  $H_{agrup}$  são da forma  $|x_1^1, x_1^2, x_2^1, x_2^2, x_3^1, x_3^2, a_1^1, a_1^2, a_2^1, a_2^2\rangle$ , como discutido na Seção 6.4. Como sabemos as soluções ótimas por análise da matriz  $\lambda$ , sabemos também as bitstrings correspondentes aos estados nos quais a melhor solução está codificada.

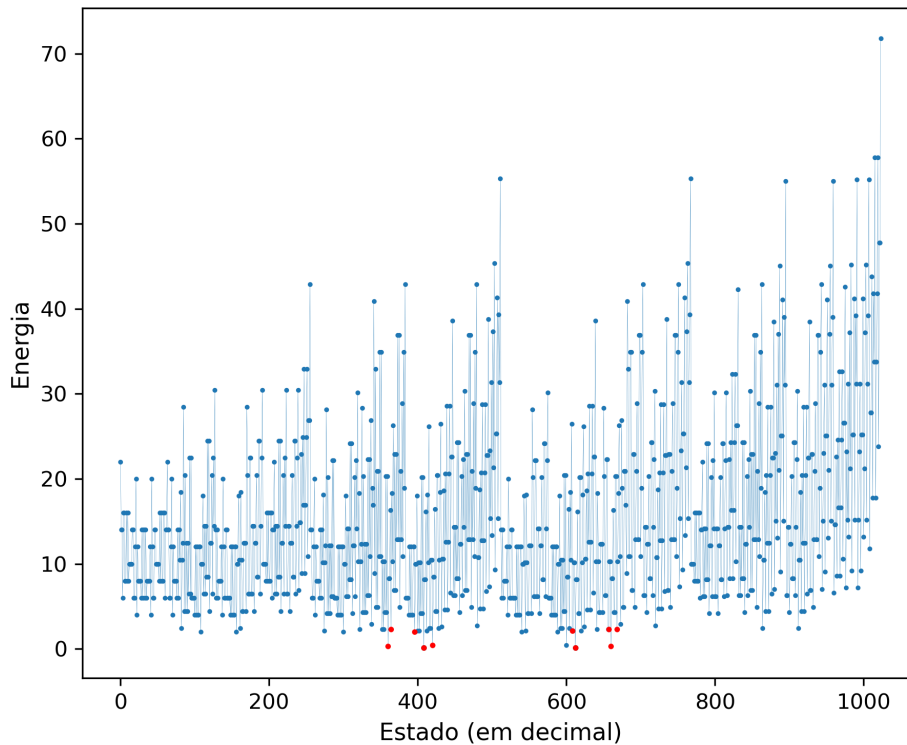


Figura 8 – Espectro do Hamiltoniano gerado na Fase de Agrupamento para o caso em que  $N = 3$  paletes devem ser distribuídos em  $M = 2$  prateleiras com  $L_m = 2$  espaços cada. Os pontos destacados em vermelho indicam os 10 estados de maior ocorrência em uma amostra.

Antes de prosseguirmos para o próximo exemplo, notemos que a Fase de Inserção não é aplicável aqui, uma vez que cada prateleira pode acomodar no máximo dois paletes. Nesse caso, a estratégia para inserção se resume a inserir o palete com maior saída em primeiro e o outro imediatamente depois.

Como segundo exemplo, consideramos agora a tarefa similar de inserir 4 paletes nas 2 prateleiras gravitacionais do depósito. Aqui, os custos-interpaletes são dados pelos elementos da matriz  $4 \times 4$  simétrica

$$\lambda = \begin{pmatrix} 0 & 0.4 & 0.2 & 0.8 \\ 0.4 & 0 & 0.6 & 0.1 \\ 0.2 & 0.6 & 0 & 0.8 \\ 0.8 & 0.1 & 0.8 & 0 \end{pmatrix},$$

a partir da qual se pode perceber que, ao distribuir os paletes entre as prateleiras, espera-se que o primeiro e o terceiro paletes sejam conjuntamente alocados na prateleira A enquanto o segundo e o quarto paletes sejam conjuntamente alocados na prateleira B, uma vez que  $\lambda_{13} < \lambda_{12} < \lambda_{14}$  e  $\lambda_{24} < \lambda_{23} < \lambda_{21}$ . Como anteriormente, uma vez que as prateleiras são indistinguíveis ( $A \rightleftharpoons B$ ), existem duas soluções ótimas para a configuração.

Na Fase de Agrupamento, o hamiltoniano expresso pela eq. (6.17) foi implementado no *quantum annealer Advantage 4.1* para o caso em que queremos distribuir eficientemente

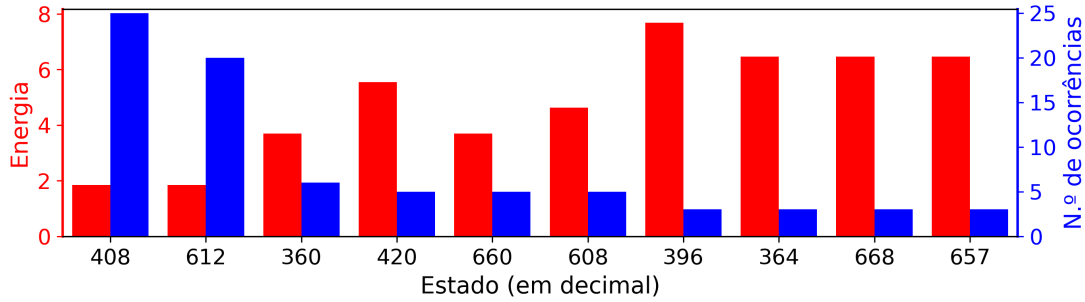


Figura 9 – Gráfico de energia (barra vermelha) e número de ocorrências (barra azul) dos 10 estados mais medidos na amostra. Os dois estados mais presentes na amostra, de *bitstrings* (em decimal) 408 e 612, correspondem ao estado fundamental degenerado de  $H_{agrup}$ , observados 25 e 20 vezes, respectivamente, com energia igual a 1.847521.

$N = 4$  paletes, com parâmetros de correspondência dados pela matriz  $\lambda$ , em  $M = 2$  prateleiras com  $L_m = 2$  espaços disponíveis em cada. As soluções esperadas, obtidas da análise da matriz  $\lambda$ , estão codificadas nas *bitstrings* 011001100000 (1632, em decimal) e 100110010000 (2448, em decimal).

Os resultados obtidos mostram que, para uma amostra com 100 *shots*, os estados fundamentais do hamiltoniano aparecem em média 48,28 vezes. Foram necessários 17 qubits físicos para mapear os 12 qubits lógicos necessários para implementação da tarefa na topologia do *quantum annealer*. Novamente, o tratamento estatístico foi realizado com 50 *runs*.

Nota-se que aqui, em comparação com o caso para  $N = 3$  paletes, houve um aumento médio de aproximadamente 13% no número de soluções desejadas encontradas na Fase de Agrupamento. Embora seja difícil apontar exatamente quais pontos levaram a essa melhoria, apontamos como causa a escolha dos parâmetros de penalização utilizados. Essa aparente contradição, portanto, nos leva a crer que existe um conjunto de parâmetros de penalização que permite resultados ainda melhores do que os aqui expostos. Novas considerações sobre o tema serão realizadas no fechamento dessa dissertação, ao apontar direcionamentos para os futuros trabalhos que se dediquem ao tema.

Ainda, nas Figuras 10 e 11, uma das 50 amostras com 100 *shots* é analisada. Na Figura 10, o espectro do hamiltoniano é representado graficamente, com os pontos em vermelho indicando os dez estados de maior ocorrência na amostra. Novamente, nota-se que todos os pontos representam estados de mínimos locais (ou globais). Na Figura 11, a energia e o número de ocorrências desses estados são apresentados, mostrando que os dois estados fundamentais que codificam a solução ótima foram encontrados em 39% dos *shots*.

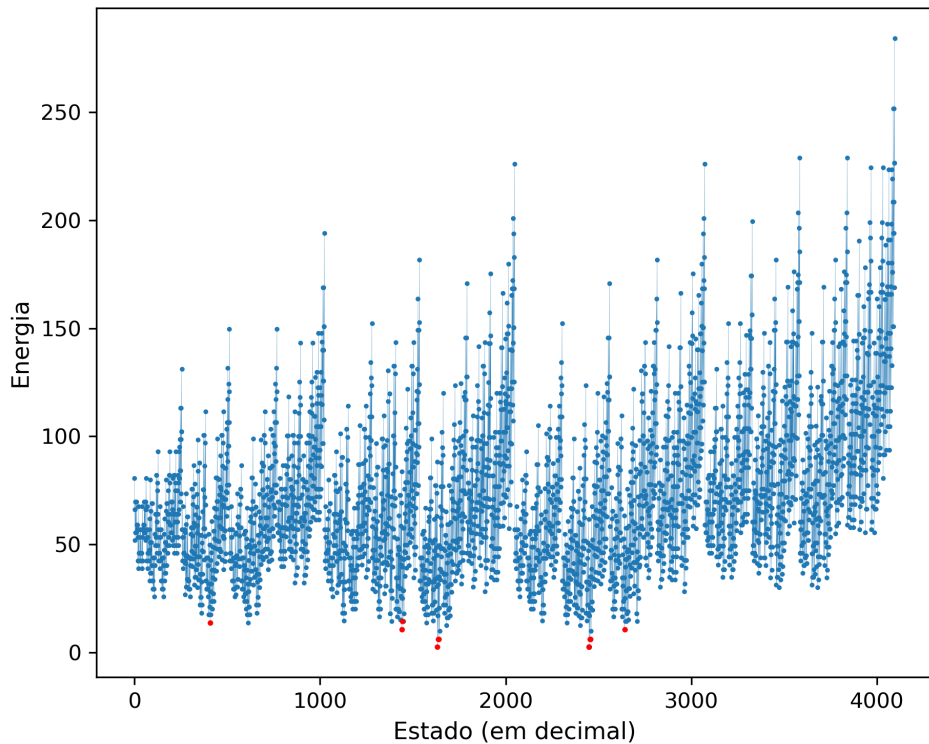


Figura 10 – Espectro do Hamiltoniano gerado na Fase de Agrupamento para o caso em que  $N = 4$  paletes devem ser distribuídos em  $M = 2$  prateleiras com  $L_m = 2$  espaços cada. Os pontos destacados em vermelho indicam os 10 estados de maior ocorrência em uma amostra.

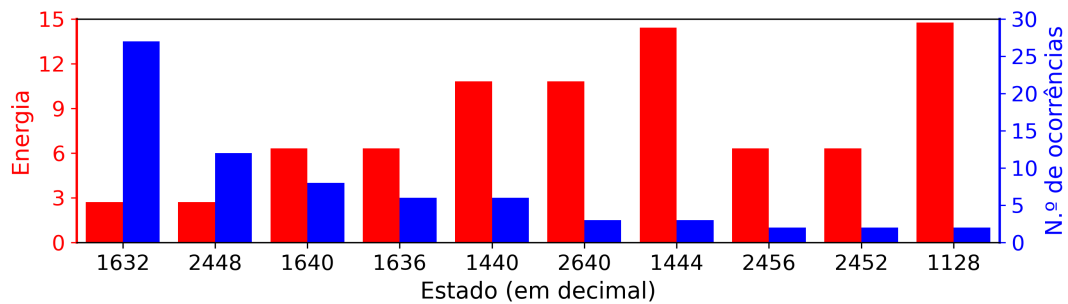


Figura 11 – Gráfico de energia (barra vermelha) e número de ocorrências (barra azul) dos 10 estados mais medidos na amostra. Os dois estados mais presentes na amostra, de *bitstrings* (em decimal) 1632 e 2448, correspondem ao estado fundamental degenerado de  $H_{agrup}$ , observados 27 e 12 vezes, respectivamente, com energia igual a 2.7.

### 6.5.2 Depósito com Três Prateleiras Gravitacionais, com $L_m = 3$

Agora, com o propósito de testar o algoritmo em um conjunto mais complexo de instâncias, vamos considerar a tarefa de alocar um conjunto de até 9 paletes em um depósito com 3 prateleiras gravitacionais, com 3 espaços cada.

Para essa tarefa, consideraremos os custos-interpaletes  $\lambda_{\alpha\beta}$  dados pelos elementos da matriz  $9 \times 9$  simétrica

$$\lambda = \begin{pmatrix} 0 & 0.0763 & 0.0763 & 0.1 & 0.125 & 0 & 0.05 & 0.0238 & 0.125 \\ 0.0763 & 0 & 0 & 0.0789 & 0.1052 & 0.0763 & 0.0763 & 0.0476 & 0.1673 \\ 0.0763 & 0 & 0 & 0.0789 & 0.1052 & 0.0763 & 0.0763 & 0.045 & 0.1673 \\ 0.1 & 0.0789 & 0.0789 & 0 & 0.0312 & 0.1 & 0.1 & 0.1190 & 0.1964 \\ 0.125 & 0.1052 & 0.1052 & 0.0312 & 0 & 0.125 & 0.125 & 0.1428 & 0.1714 \\ 0 & 0.0763 & 0.0763 & 0.1 & 0.125 & 0 & 0 & 0.0238 & 0.125 \\ 0.05 & 0.0763 & 0.0763 & 0.1 & 0.125 & 0 & 0 & 0.0238 & 0.125 \\ 0.0238 & 0.0476 & 0.045 & 0.119 & 0.1428 & 0.0238 & 0.0238 & 0 & 0.1428 \\ 0.125 & 0.1673 & 0.1673 & 0.1964 & 0.1714 & 0.125 & 0.125 & 0.1428 & 0 \end{pmatrix}.$$

Diferentemente das matrizes utilizadas nos exemplos passados, esses parâmetros foram gerados via *machine learning* a partir dos registros das demandas da linha de produção, como discutido na Seção 6.2. Nesse sentido, a matriz  $\lambda$  é um recorte dos dados reais do depósito da fábrica parceira desse projeto.

#### Fase de Agrupamento

Na Fase de Agrupamento, implementamos o hamiltoniano da eq. (6.17) no *quantum annealer Advantage 4.1* para os casos em que queremos distribuir eficientemente de  $N = 4$  até  $N = 9$  paletes em  $M = 3$  prateleiras com  $L_m = 3$  espaços disponíveis em cada, com custos-interpaletes dados pelas submatrizes  $N \times N$  da matriz  $\lambda$ .

Aqui, em vez de buscarmos determinar quantas vezes a melhor solução (codificada no estado fundamental com sêxtupla degenerescência<sup>8</sup> de  $H_{agrup}$ ) é encontrada dentro de um conjunto de amostras com um número específico de *shots* (como fizemos nos exemplos anteriores), optamos por determinar quantos *shots* são necessários para que a melhor solução seja encontrada pelo menos uma vez. A abordagem que optamos permite determinar qual a menor quantidade de *shots* (e, portanto, de tempo) necessária para alcançar a solução ótima da tarefa.

Para os casos com 4, 5 e 6 paletes, como demonstrado no Painel 12, são necessários 35, 75 e 175 *shots* para alcançar a solução ótima, respectivamente. Na construção dos gráficos, realizamos 50 *runs* para cada valor de *Número de Shots*, salvando a menor energia de cada *run*, e calculamos uma média sobre esses valores, que chamamos de *Energia Média*.

<sup>8</sup> Como as três prateleiras são indistinguíveis, existem seis configurações ótimas para cada caso.

As soluções encontradas para essas instâncias são descritas na Tabela 1 e concordam com as soluções encontradas classicamente, obtidas a partir de uma busca exaustiva no espectro do hamiltoniano da tarefa.

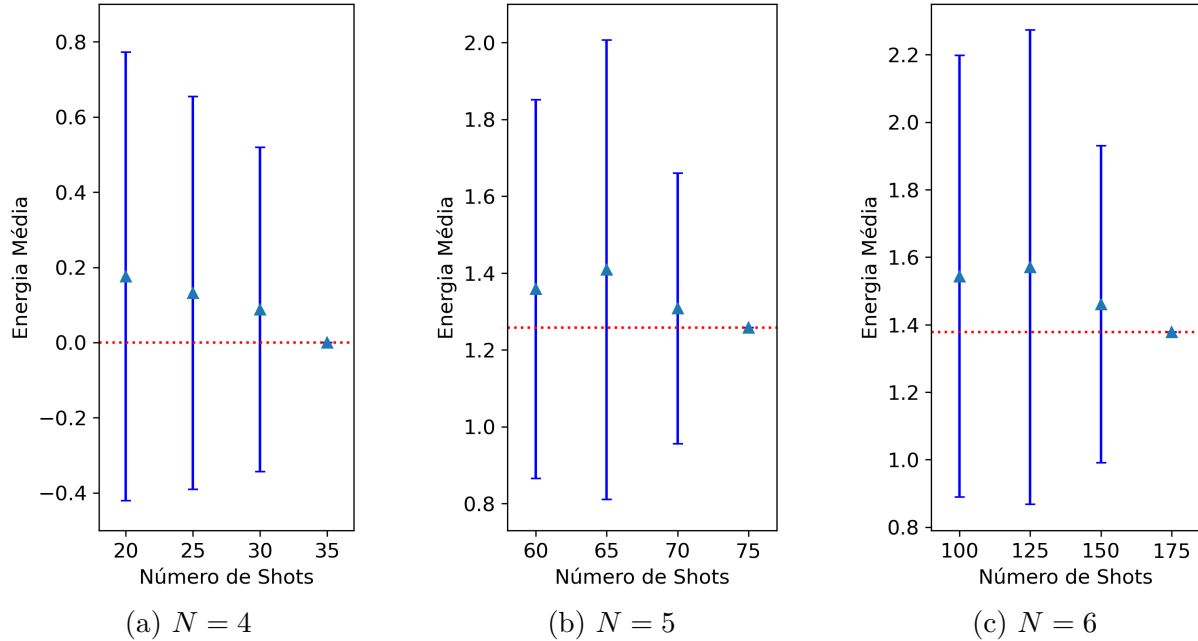


Figura 12 – No Painel, determinamos o número de *shots* necessários para alcançar a solução ótima da Fase de Agrupamento em (a) para  $N = 4$ , em (b) para  $N = 5$  e em (c) para  $N = 6$ . A linha pontilhada em vermelho representa a energia do estado fundamental do Hamiltoniano de Agrupamento, obtida a partir de uma busca exaustiva no espectro do hamiltoniano.

Número de Paletes	Solução Obtida	Energia da Solução	É a melhor?
4	Prateleira A: 1 Prateleira B: 2 e 3 Prateleira C: 4	0	✓
5	Prateleira A: 1 Prateleira B: 2 e 3 Prateleira C: 4 e 5	1.257	✓
6	Prateleira A: 1 e 6 Prateleira B: 2 e 3 Prateleira C: 4 e 5	1.377	✓

Tabela 1 – Soluções encontradas para a Fase de Agrupamento nos casos em que se quer distribuir 4, 5 e 6 paletes em 3 prateleiras gravitacionais de um depósito, cada uma com 3 espaços disponíveis.

De forma análoga, para os casos com 7, 8 e 9 paletes, como demonstrado no Painel 13, determinamos que são necessários 200, 575 e 3400 *shots* para alcançar a solução ótima, respectivamente. As soluções encontradas para essas instâncias são descritas na Tabela 2.

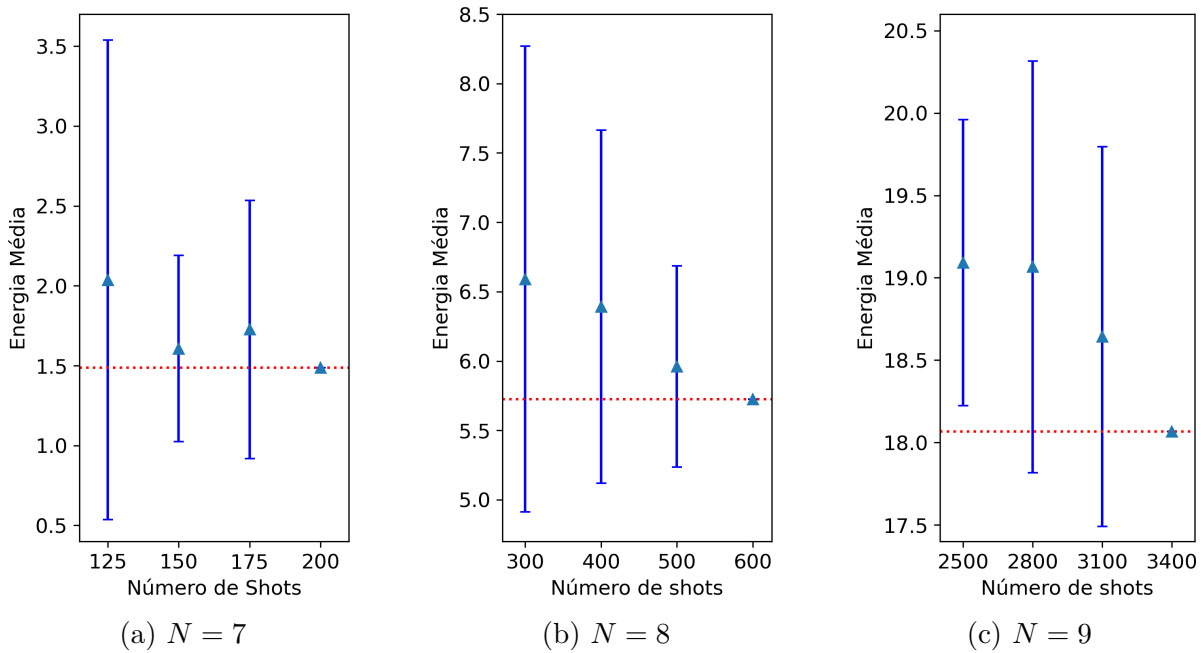


Figura 13 – No Painel, determinamos o número de *shots* necessários para alcançar a solução ótima da Fase de Agrupamento em (a) para  $N = 7$ , em (b) para  $N = 8$  e em (c) para  $N = 9$ . A linha pontilhada em vermelho representa a energia do estado fundamental do Hamiltoniano de Agrupamento, obtida a partir de uma busca exaustiva no espectro do hamiltoniano.

Número de Paletes	Solução Obtida	Energia da Solução	É a melhor?
7	Prateleira A: 1, 6 e 7 Prateleira B: 2 e 3 Prateleira C: 4 e 5	1.488	✓
8	Prateleira A: 1, 6 e 7 Prateleira B: 2, 3 e 8 Prateleira C: 4 e 5	5.724	✓
9	Prateleira A: 1, 6, 7 Prateleira B: 2, 3 e 8 Prateleira C: 4, 5, 9	18.066	✓

Tabela 2 – Soluções encontradas para a Fase de Agrupamento nos casos em que se quer distribuir 7, 8 e 9 paletes em 3 prateleiras gravitacionais de um depósito, cada uma com 3 espaços disponíveis.

Aqui, notamos que ocorre um crescimento vertiginoso do número de *shots* necessários para alcançar a solução ótima no caso com 9 paletes. Novamente, apontamos que a causa desse crescimento brusco pode ser justificada pela escolha dos parâmetros de penalização. Uma alternativa para o método de escolha dos parâmetros será discutida nas nossas considerações finais.

Por fim, considerando os resultados expostos nos Painéis 12 e 13 acerca da implementação da Fase de Agrupamento para os casos em que se quer distribuir de 4 a 9 paletes

nas prateleiras desse depósito, apresentamos na Figura 14 dois aspectos de importante análise:

- Na Figura 14(a), apresentamos o tempo médio necessário para implementar a Fase de Agrupamento no *quantum annealer Advantage 4.1*, alcançando a melhor solução ao utilizar o número de *shots* determinado. As barras de erro estão relacionadas ao tempo de leitura e pós-processamento necessário para cada *shot* dentro de uma *run*, que varia com o número de qubits físicos utilizados e/ou da seção da topologia em que os qubits lógicos foram mapeados [99].

Os resultados demonstram que o tempo gasto para determinar a melhor solução nos casos em que se busca distribuir eficientemente entre  $N = 4$  e  $N = 8$  paletes nas prateleiras gravitacionais do depósito não ultrapassa 0.1 segundo, mas ultrapassa meio segundo para o caso com 9 paletes.

- Na Figura 14(b), comparamos o número de qubits físicos com o número de qubits lógicos necessários para implementação da Fase de Agrupamento para cada número de paletes. As barras de erro estão relacionadas ao fato de que o mapeamento foi executado utilizando as ferramentas nativas da *D-Wave*, as quais empregam heurísticas clássicas e, conseqüentemente, utilizam diferentes quantidades de qubits físicos para cada execução.

Como discutido anteriormente, esse crescimento ocorre devido ao número de acoplamentos necessários entre os qubits lógicos para implementação do algoritmo extrapolar o número de acoplamentos que existem entre os qubits físicos dos *quantum annealers*. Para contornar esse problema, por sua vez, qubits físicos auxiliares são utilizados para mediar o acoplamento entre diferentes células de qubits dos dispositivos, aumentando o número efetivo de acoplamentos ao preço de também aumentar o número de qubits físicos utilizados.

### Fase de Inserção

Para que a Fase de Inserção seja aplicável, é necessário que  $N_m \geq 3$ , o que ocorre para os casos com 7, 8 e 9 paletes. Para essas instâncias, implementamos o hamiltoniano expresso pela eq. (6.18) no *quantum annealer Advantage 4.1* para os grupos de paletes determinados durante a Fase de Agrupamento (indicados na Tabela 2). O parâmetro de penalização do hamiltoniano foi ajustado para  $D = N_m$ .

Os resultados obtidos em 50 *runs* com 100 *shots* cada determinaram que a configuração ótima para a sequência de inserção dos paletes em suas respectivas prateleiras

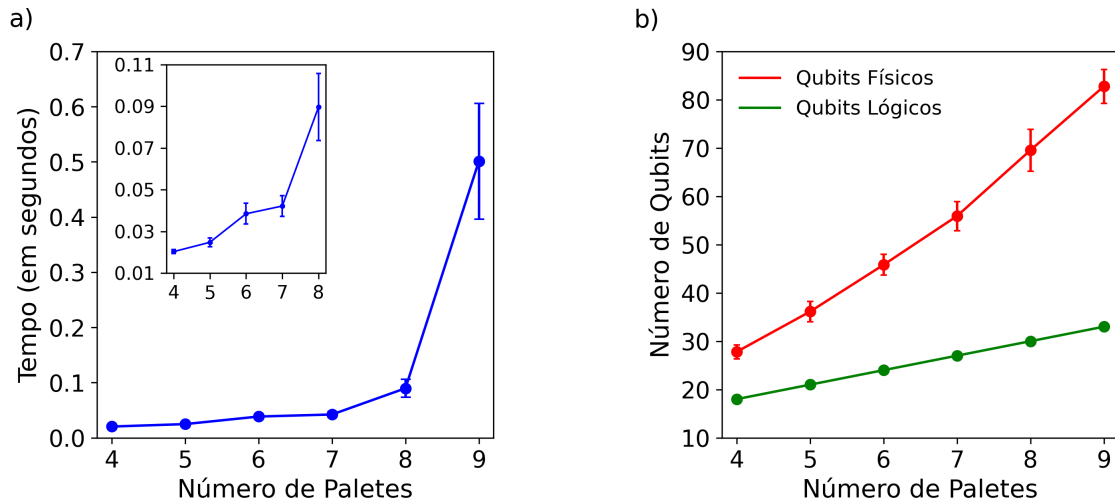


Figura 14 – Análise do tempo e do número de qubits físicos necessários para a implementação da Fase de Agrupamento no *quantum annealer Advantage 4.1* da *D-Wave* para o caso estudado. Em (a) o tempo necessário utilizando o número de shots determinado para alcançar a solução ótima pelo menos uma vez e em (b) o número de qubits físicos utilizados em comparação com o número de qubits lógicos para cada instância.

gravitacionais é **A)**  $1 \rightarrow 6 \rightarrow 7^9$ , **B)**  $2 \rightarrow 3 \rightarrow 8$  e **C)**  $4 \rightarrow 5 \rightarrow 9$ . Os estados fundamentais foram encontrados em média **A)** 36,12, **B)** 30 e **C)** 26,44 vezes nas amostras.

Embora as instâncias abordadas acima sejam simples e possam ser resolvidas classicamente sem quaisquer dificuldades, é importante lembrar que a tarefa é mapeada no Problema do Caixeiro Viajante, cuja complexidade foi discutida em capítulos anteriores e torna-se de difícil resolução para instâncias maiores e mais complexas. Ainda, uma vez que reconhecemos que grande parte dos depósitos possui  $N_m$  da ordem de algumas dezenas, é viável considerar implementações híbridas do algoritmo, abordando a Fase de Inserção classicamente. Aqui, no entanto, ressaltamos que preferimos uma abordagem totalmente quântica para que, em acordo com o objetivo dessa exposição, fosse possível verificar o funcionamento do algoritmo em duas fases nos *quantum annealers* da *D-Wave*.

<sup>9</sup> Na notação: Para a prateleira A, o sétimo paleta deve ser inserido em primeiro lugar, o sexto paleta em segundo lugar e por último o primeiro paleta.

## 7 Conclusões e Próximos Passos

Nessa dissertação, tivemos como principal objetivo desenvolver um algoritmo quântico para o Problema de Gerenciamento de Estoque em Computação Quântica Adiabática, adaptado para fábricas que utilizam prateleiras gravitacionais como sistema padrão de armazenamento. Para isso, elaboramos uma estratégia em duas fases, nomeadas como Fase de Agrupamento e Fase de Inserção.

Na Fase de Agrupamento, busca-se determinar a melhor forma de distribuir um conjunto de  $N$  paletes em  $M$  prateleiras gravitacionais, cada uma com  $P_m$  paletes previamente alocados. Para definir um critério de otimização que permitisse determinar em quais prateleiras cada palete deveria ser alocado (e, portanto, quais paletes deveriam ser alocados nas mesmas prateleiras), foram utilizados parâmetros de correspondência (gerados via *machine learning* a partir dos registros das demandas da linha de produção da fábrica) que medem quão recorrentemente diferentes insumos são conjuntamente requisitados.

Ao fim da etapa, obtém-se como resultado a melhor forma de distribuir os  $N$  paletes em  $M$  grupos, cada um contendo  $N_m$  paletes, com o  $m$ -ésimo grupo correspondendo ao conjunto de paletes destinado a ser alocado na  $m$ -ésima prateleira, de modo que se maximize a correspondência entre os paletes (novos e previamente alocados) nas mesmas prateleiras, reduzindo assim o número de operações nas futuras retiradas do depósito.

Na Fase de Inserção, para cada uma das  $M$  prateleiras gravitacionais, busca-se determinar a melhor sequência para inserir os  $M$  conjuntos de paletes em suas respectivas prateleiras, seguindo as restrições trazidas pelo princípio de gerenciamento FIFO. Para isso, foi implementada uma estratégia inspirada no Problema do Caixeiro Viajante de tal forma que os paletes sejam inseridos na sequência que maximize a correspondência dos paletes inseridos em posições adjacentes.

Para implementar a estratégia elaborada no modelo de Computação Quântica Adiabática, codificamos o espaço de soluções de cada fase nos autoestados de um Hamiltoniano Problema correspondente, definidos pelas eqs. (6.14) e (6.16), de tal forma que a solução ótima fosse codificada nos respectivos estados fundamentais de cada hamiltoniano. Ambos os hamiltonianos foram formulados como QUBOs, possibilitando suas implementações nos *quantum annealers* disponibilizados em nuvem pela *D-Wave Systems*. Consideramos como o algoritmo quântico desenvolvido para o Problema de Gerenciamento de Estoque (conforme estruturado nos capítulos anteriores) a aplicação sequencial da Fase de Agrupamento e da Fase de Inserção.

Para atestar seu funcionamento, aplicamos o algoritmo para um conjunto de instâncias simples no *quantum annealer Advantage 4.1*. Nesse conjunto de instâncias,

denotamos como  $L_m$  a capacidade máxima de paletes que a  $m$ -ésima prateleira pode acomodar. Além disso, consideramos prateleiras inicialmente vazias, isto é, com  $P_m = 0$ , o que permitiu reduzir os Hamiltonianos Problemas para as formas apresentadas nas eqs. (6.17) e (6.18).

Embora simples, essas instâncias mantêm características importantes dos hamiltonianos originais, como o número de qubits lógicos empregados em ambas as fases. Portanto, ao implementar o algoritmo para abordar esses casos, pudemos determinar como as limitações do atual estado da arte dos *quantum annealers* afetam os resultados obtidos. Ainda, em uma interpretação que permite futuras comparações com implementações em que  $P_m \neq 0$ , reduzir os Hamiltonianos Problemas para as formas apresentadas nas eqs. (6.17) e (6.18) implica uma estratégia mais simples onde não são consideradas as correspondências também entre os paletes previamente alocados na distribuição de novos paletes que chegam ao depósito.

Um ponto que deve ser destacado nas nossas implementações é que, para os hamiltonianos utilizados para abordar essas instâncias, encontramos valores numéricos para os parâmetros de penalização que encontraram consistentemente a melhor solução para os casos mais simples com dezenas e os casos mais complexos com centenas de *shots*. No entanto, foram necessários 3400 *shots* para encontrar a melhor solução para o caso em que aplicamos a Fase de Agrupamento à tarefa de distribuir  $N = 9$  paletes em  $M = 3$  prateleiras com  $L_m = 3$ . Esse número é aproximadamente 5,5 vezes maior do que aquele utilizado para encontrar a solução ótima para o caso com  $N = 8$  paletes.

Esse aumento significativo no número de *shots* para o caso com 9 paletes pode ser atribuído à escolha dos parâmetros de penalização. A esse respeito, concluímos que o método geral de determinar um conjunto específico de parâmetros para cada instância pode não ser eficaz em todos os cenários. Nesse sentido, como alternativa para futuras investigações, consideramos a possibilidade de adotar uma abordagem que consista em fixar um intervalo de tempo, como, por exemplo, 1 segundo, e realizar uma varredura dentro de uma faixa de valores para os parâmetros de penalização, executando um número pré-determinado de *shots* para cada conjunto de valores.

Outra questão ainda não abordada é qual a instância mais complexa que pode ser abordada nos *quantum annealers* da *D-Wave*, considerando as atuais limitações desses dispositivos. No que se refere a esse ponto, conduzimos testes que evidenciam que a instância mais complexa que pode ser abordada no *quantum annealer Advantage 4.1* é aquela em que uma coleção com até 36 paletes devem ser distribuídos em 6 prateleiras gravitacionais, cada uma com 6 espaços disponíveis. Essa limitação está diretamente relacionada a quantidade limitada de qubits lógicos nas topologias desses dispositivos, o que, por sua vez, se traduz no aumento do número de qubits físicos necessários.

De forma geral, concluímos que os resultados expostos nessa dissertação atendem

---

aos objetivos inicialmente propostos e estabelecem uma base sólida para futuros trabalhos. Tais investigações estão em andamento, envolvendo não apenas o autor, mas também outros membros do grupo de pesquisa. Pontuamos aqui as principais linhas que estão sendo atualmente investigadas:

- Com o objetivo de determinar o ganho real do algoritmo, estão sendo realizados testes que permitem comparar o número de reinserções necessárias no gerenciamento de estoque utilizando a estratégia elaborada a abordagens mais simples, como a inserção aleatória dos paletes nas prateleiras e a inserção seguindo algoritmos de recomendação gerados por *machine learning*.
- Com o objetivo de avaliar a eficiência do algoritmo quando implementado em *Quantum Annealing*, estão sendo realizadas implementações dos hamiltonianos desenvolvidos utilizando abordagens clássicas, como o *Simulated Annealing*. Essa análise comparativa é fundamental para compreender as possíveis vantagens e desvantagens obtidas pela CQA para a tarefa.
- Com o objetivo de analisar possíveis vantagens de outras abordagens quânticas, o algoritmo está sendo implementado em QAOA<sup>1</sup> [100], utilizando sistemas baseados em íons aprisionados, em uma colaboração com a empresa austríaca *Alpine Quantum Technologies* (AQT) [101].
- Por fim, com o objetivo de abordar instâncias maiores e mais complexas, estão sendo exploradas a implementação do algoritmo em modelos de computação híbrida, como o CQM [102] da *D-Wave Systems*.

---

<sup>1</sup> *Quantum Approximate Optimization Algorithm*. Em tradução literal, Algoritmo de Otimização Quântica Aproximada.



# Referências

- [1] Kyle Erwin and Andries Engelbrecht. Meta-heuristics for portfolio optimization. *Soft Computing*, April 2023. Citado na página 17.
- [2] Frederick S Hillier. *Introduction to operations research*. February 2020. Citado na página 17.
- [3] David L. Applegate, Robert E. Bixby, Vašek Chvátal, and William J. Cook. 2007. Citado 3 vezes nas páginas 17, 25 e 26.
- [4] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer Berlin Heidelberg, 2004. Citado 4 vezes nas páginas 17, 25, 27 e 28.
- [5] Gilbert Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, June 1992. Citado 4 vezes nas páginas 17, 25, 27 e 30.
- [6] Pavlo Krokhmal, Tanislav Uryasev, and Jonas Palmquist. Portfolio optimization with conditional value-at-risk objective and constraints. *The Journal of Risk*, 4(2):43–68, March 2001. Citado 2 vezes nas páginas 17 e 25.
- [7] Arnab Das and Bikas K. Chakrabarti. Colloquium: Quantum annealing and analog quantum computation. *Reviews of Modern Physics*, 80(3):1061–1081, September 2008. Citado na página 17.
- [8] E. J. Crosson and D. A. Lidar. Prospects for quantum enhancement with diabatic quantum annealing. *Nature Reviews Physics*, 3(7):466–489, May 2021. Citado na página 17.
- [9] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, June 1982. Citado na página 18.
- [10] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994. Citado na página 18.
- [11] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*. ACM Press, 1996. Citado na página 18.
- [12] Amanda Hiley and Bryant A. Julstrom. The quadratic multiple knapsack problem and three heuristic approaches to it. In *Proceedings of the 8th annual conference on*

- Genetic and evolutionary computation*. ACM, July 2006. Citado 2 vezes nas páginas 18 e 54.
- [13] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, October 2019. Citado na página 18.
- [14] Feng Pan, Keyang Chen, and Pan Zhang. Solving the sampling problem of the sycamore quantum circuits. *Physical Review Letters*, 129(9), August 2022. Citado na página 18.
- [15] Peter Nimbe, Benjamin Asubam Weyori, and Adebayo Felix Adekoya. Models in quantum computing: a systematic review. *Quantum Information Processing*, 20(2), February 2021. Citado na página 18.
- [16] Tameem Albash and Daniel A. Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1), January 2018. Citado 3 vezes nas páginas 18, 31 e 34.
- [17] Paulo J. P. Souza, Taysa M. Mendonça, Estêvão V. B. de Oliveira, and Celso J. Villas-Boas. Computação Quântica Adiabática: Do Teorema Adiabático ao Computador da *D-Wave*. *Revista Brasileira de Ensino de Física*, 43, 2021. Citado 6 vezes nas páginas 18, 31, 33, 34, 37 e 39.
- [18] D-Wave: Error Sources for Problem Representation. [https://docs.dwavesys.com/docs/latest/c\\_qpu\\_ice.html](https://docs.dwavesys.com/docs/latest/c_qpu_ice.html). Acesso em 30/09/2023. Citado 2 vezes nas páginas 18 e 35.

- [19] D-Wave: Error Sources for Problem Representation. [https://docs.dwavesys.com/docs/latest/c\\_qpu\\_ice.html](https://docs.dwavesys.com/docs/latest/c_qpu_ice.html). Acesso em 30/09/2023. Citado 2 vezes nas páginas 18 e 35.
- [20] Página Inicial da D-Wave Systems. <https://www.dwavesys.com/>. Acesso em 30/09/2023. Citado 2 vezes nas páginas 19 e 31.
- [21] Catherine C. McGeoch. *Adiabatic Quantum Computation and Quantum Annealing*. Springer International Publishing, 2014. Citado 2 vezes nas páginas 19 e 33.
- [22] Siddharth Jain. Solving the traveling salesman problem on the d-wave quantum computer. *Frontiers in Physics*, 9, November 2021. Citado 2 vezes nas páginas 19 e 31.
- [23] Alberto Ceselli and Marco Premoli. On good encodings for quantum annealer and digital optimization solvers. *Scientific Reports*, 13(1), April 2023. Citado 2 vezes nas páginas 19 e 31.
- [24] Sebastian Feld, Christoph Roch, Thomas Gabor, Christian Seidel, Florian Neukart, Isabella Galter, Wolfgang Mauerer, and Claudia Linnhoff-Popien. A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. *Frontiers in ICT*, 6, June 2019. Citado 3 vezes nas páginas 19, 31 e 45.
- [25] Augusto Da Cunha Reis, Cristina Gomes de Souza, Nayara Nogueira da Costa, Gustavo Henrique Cordeiro Stender, Pedro Senna Vieira, and Nélio Domingues Pizzolato. Warehouse design: a systematic literature review. *Brazilian Journal of Operations & Production Management*, 14(4):542–555, December 2017. Citado 2 vezes nas páginas 19 e 47.
- [26] LinkedIn - Flow Rack na Logística e Como Funciona. <https://www.dwavesys.com/>. Acesso em 30/09/2023. Citado 2 vezes nas páginas 19 e 47.
- [27] Forbes: What Is The FIFO Method? FIFO Inventory Guide. <https://www.forbes.com/advisor/business/fifo-method/>. Acesso em 30/09/2023. Citado 2 vezes nas páginas 19 e 48.
- [28] Andrew Lucas. Ising formulations of many NP problems. *Frontiers in Physics*, 2, 2014. Citado 4 vezes nas páginas 21, 39, 44 e 45.
- [29] Mark M Meerschaert. *Mathematical Modeling*. Academic Press, San Diego, CA, 4 edition, January 2013. Citado na página 25.
- [30] Bernhard Korte and Jens Vygen. *Combinatorial Optimization*. Springer Berlin Heidelberg, 2018. Citado na página 25.

- [31] Juris Hartmanis. Computers and intractability: A guide to the theory of NP-completeness (michael r. garey and david s. johnson). *SIAM Review*, 24(1):90–91, January 1982. Citado na página 25.
- [32] Hajar Lamghari Elidrissi, Ahmed Nait-Sidi-Moh, and Abdelouahed Tajer. Knapsack problem-based control approach for traffic signal management at urban intersections: Increasing smooth traffic flows and reducing environmental impact. *Ecological Complexity*, 44:100878, December 2020. Citado na página 25.
- [33] Karla L. Hoffman, Manfred Padberg, and Giovanni Rinaldi. Traveling salesman problem. In *Encyclopedia of Operations Research and Management Science*, pages 1573–1578. Springer US, 2013. Citado na página 25.
- [34] Valentina Cacchiani, Manuel Iori, Alberto Locatelli, and Silvano Martello. Knapsack problems — an overview of recent advances. part i: Single knapsack problems. *Computers & Operations Research*, 143:105692, July 2022. Citado 2 vezes nas páginas 25 e 28.
- [35] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, April 2011. Citado na página 25.
- [36] Paolo Toth and Daniele Vigo. 1. an overview of vehicle routing problems. In *The Vehicle Routing Problem*, pages 1–26. Society for Industrial and Applied Mathematics, January 2002. Citado na página 26.
- [37] Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM*, 9(1):61–63, January 1962. Citado 2 vezes nas páginas 26 e 27.
- [38] Reinhard Diestel. *Graph Theory*. Springer Berlin Heidelberg, 2017. Citado na página 27.
- [39] Abraham P. Punnen. The traveling salesman problem: Applications, formulations and variations. In *Combinatorial Optimization*, pages 1–28. Springer US, 2007. Citado na página 27.
- [40] Norman Biggs. The travelling salesman problem: A guided tour of combinatorial optimization. *Bulletin of the London Mathematical Society*, 18(5):514–515, September 1986. Citado na página 27.
- [41] Xavier Bresson and Thomas Laurent. The transformer network for the traveling salesman problem, 2021. Citado na página 27.

- [42] Tariq Alzyadat, Mohammad Yamin, and Girija Chetty. Genetic algorithms for the travelling salesman problem: a crossover comparison. *International Journal of Information Technology*, 12(1):209–213, November 2019. Citado na página 27.
- [43] Silvano Martello and Paolo Toth. *Knapsack problems*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Chichester, England, August 1990. Citado na página 27.
- [44] Silvano Martello, David Pisinger, and Paolo Toth. Dynamic programming and strong bounds for the 0-1 knapsack problem. *Management Science*, 45(3):414–424, March 1999. Citado na página 27.
- [45] Valentina Cacchiani, Manuel Iori, Alberto Locatelli, and Silvano Martello. Knapsack problems — an overview of recent advances. part i: Single knapsack problems. *Computers & Operations Research*, 143:105692, July 2022. Citado 2 vezes nas páginas 28 e 54.
- [46] David Pisinger and Stefan Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, August 2007. Citado 2 vezes nas páginas 28 e 54.
- [47] Andrea Bettinelli, Valentina Cacchiani, and Enrico Malaguti. A branch-and-bound algorithm for the knapsack problem with conflict graph. *INFORMS Journal on Computing*, 29(3):457–473, August 2017. Citado na página 29.
- [48] Absalom E. Ezugwu, Verosha Pillay, Divyan Hirasen, Kershen Sivanarain, and Melvin Govender. A comparative study of meta-heuristic optimization algorithms for 0 – 1 knapsack problem: Some initial results. *IEEE Access*, 7:43979–44001, 2019. Citado na página 29.
- [49] Yao-Hsin Chou, Yi-Jyuan Yang, and Chia-Hui Chiu. Classical and quantum-inspired tabu search for solving 0/1 knapsack problem. In *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1364–1369, 2011. Citado na página 29.
- [50] Gilbert Laporte and Frédéric Semet. 5. classical heuristics for the capacitated VRP. In *The Vehicle Routing Problem*, pages 109–128. Society for Industrial and Applied Mathematics, January 2002. Citado 2 vezes nas páginas 29 e 30.
- [51] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, October 1959. Citado na página 29.
- [52] Bruce Golden, S. Raghavan, and Edward Wasil, editors. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer US, 2008. Citado na página 29.

- 
- [53] Billy E. Gillett and Leland R. Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349, April 1974. Citado na página 30.
- [54] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, England, December 2010. Citado na página 31.
- [55] Jack D. Hidary. *Quantum Computing: An Applied Approach*. Springer International Publishing, 2019. Citado na página 31.
- [56] Hirotaka Irie, Goragot Wongpaisarnsin, Masayoshi Terabe, Akira Miki, and Shini-chirou Taguchi. Quantum annealing of vehicle routing problem with time, state and capacity. In *Quantum Technology and Optimization Problems*, pages 145–156. Springer International Publishing, 2019. Citado 2 vezes nas páginas 31 e 45.
- [57] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983. Citado na página 33.
- [58] Peter J. M. van Laarhoven and Emile H. L. Aarts. *Simulated Annealing: Theory and Applications*. Springer Netherlands, 1987. Citado na página 33.
- [59] Nazmul Siddique and Hojjat Adeli. Simulated annealing, its variants and engineering applications. *International Journal on Artificial Intelligence Tools*, 25(06):1630001, October 2016. Citado na página 33.
- [60] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution, 2000. Citado na página 34.
- [61] M. Born and V. Fock. Beweis des adiabatensatzes. *Zeitschrift für Physik*, 51(3-4):165–180, March 1928. Citado na página 34.
- [62] David J Griffiths and Darrell F Schroeter. *Introduction to Quantum Mechanics*. Cambridge University Press, Cambridge, England, 3 edition, August 2018. Citado na página 34.
- [63] Daniel Comparat. General conditions for quantum adiabatic evolution. *Phys. Rev. A*, 80:012106, Jul 2009. Citado na página 35.
- [64] Zhen-Yu Wang and Martin B. Plenio. Necessary and sufficient condition for quantum adiabatic evolution by unitary control fields. *Physical Review A*, 93(5), May 2016. Citado na página 35.
- [65] D-Wave: What is Quantum Annealing? [https://docs.dwavesys.com/docs/latest/c\\_gs\\_2.html](https://docs.dwavesys.com/docs/latest/c_gs_2.html). Acesso em 30/09/2023. Citado na página 35.

- [66] Atanu Rajak, Sei Suzuki, Amit Dutta, and Bikas K. Chakrabarti. Quantum annealing: an overview. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 381(2241), December 2022. Citado na página 35.
- [67] AWS Marketplace: About D-Wave Commercial Inc. <https://aws.amazon.com/marketplace/seller-profile?id=aa4f61b5-7b1f-4984-9175-1a636281d694>. Acesso em 30/09/2023. Citado na página 36.
- [68] D-Wave: When Was D-Wave Founded? <https://support.dwavesys.com/hc/en-us/articles/360009865073>. Acesso em 30/09/2023. Citado na página 36.
- [69] Página Inicial: Leap. <https://cloud.dwavesys.com/leap/>. Acesso em 30/09/2023. Citado na página 36.
- [70] The Advantage System: Performance Update. [https://www.dwavequantum.com/media/kjtlcemb/14-1054a-a\\_advantage\\_system\\_performance\\_update.pdf](https://www.dwavequantum.com/media/kjtlcemb/14-1054a-a_advantage_system_performance_update.pdf). Acesso em 30/09/2023. Citado na página 37.
- [71] D. Drung, C. Abmann, J. Beyer, A. Kirste, M. Peters, F. Ruede, and Th. Schurig. Highly sensitive and easy-to-use squid sensors. *IEEE Transactions on Applied Superconductivity*, 17(2):699–704, 2007. Citado na página 37.
- [72] Zhengbing Bian, Fabián A. Chudak, William G. Macready, and Geordie Rose. The ising model : teaching an old problem new tricks. 2010. Citado na página 37.
- [73] D-Wave: What is Quantum Annealing? [https://docs.dwavesys.com/docs/latest/c\\_gs\\_2.html](https://docs.dwavesys.com/docs/latest/c_gs_2.html). Acesso em 30/09/2023. Citado na página 37.
- [74] D-Wave: Problem Formulation Guide. <https://www.dwavesys.com/media/bu0lh5ee/problem-formulation-guide-2022-01-10.pdf>. Acesso em 30/09/2023. Citado na página 37.
- [75] Fred Glover, Gary Kochenberger, and Yu Du. A tutorial on formulating and using qubo models, 2019. Citado na página 38.
- [76] D-Wave QPU Architecture: Topologies. [https://docs.dwavesys.com/docs/latest/c\\_gs\\_4.html](https://docs.dwavesys.com/docs/latest/c_gs_4.html). Acesso em 30/09/2023. Citado na página 38.
- [77] D-Wave: Minor Embedding. <https://docs.ocean.dwavesys.com/en/stable/concepts/embedding.html>. Acesso em 30/09/2023. Citado na página 38.
- [78] René de Koster, Tho Le-Duc, and Kees Jan Roodbergen. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2):481–501, October 2007. Citado 2 vezes nas páginas 47 e 52.

- [79] Makusee Masae, Christoph H. Glock, and Eric H. Grosse. Order picker routing in warehouses: A systematic literature review. *International Journal of Production Economics*, 224:107564, June 2020. Citado 2 vezes nas páginas 47 e 52.
- [80] MICOR: Heavy Duty Pallet Boxes. <https://www.micor.com.au/pallet-boxes/>. Acesso em 30/09/2023. Citado na página 48.
- [81] Laadur: Prateleiras de Fluxo. <https://www.laadur.ee/muuk/laosisustus/laoriiulid/labivooluriulid/>. Acesso em 30/09/2023. Citado na página 48.
- [82] Jinxiang Gu, Marc Goetschalckx, and Leon F. McGinnis. Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, 177(1):1–21, February 2007. Citado na página 52.
- [83] Letitia M. Pohl, Russell D. Meller, and Kevin R. Gue. An analysis of dual-command operations in common warehouse designs. *Transportation Research Part E: Logistics and Transportation Review*, 45(3):367–379, May 2009. Citado na página 52.
- [84] Heung Suk Hwang and Gyu Sung Cho. A performance evaluation model for order picking warehouse design. *Computers & Industrial Engineering*, 51(2):335–342, October 2006. Citado na página 52.
- [85] Pratik J. Parikh and Russell D. Meller. A travel-time model for a person-onboard order picking system. *European Journal of Operational Research*, 200(2):385–394, January 2010. Citado na página 52.
- [86] Iris F.A. Vis and Kees Jan Roodbergen. Layout and control policies for cross docking operations. *Computers & Industrial Engineering*, 61(4):911–919, November 2011. Citado na página 52.
- [87] B. Rouwenhorst, B. Reuter, V. Stockrahm, G.J. van Houtum, R.J. Mantel, and W.H.M. Zijm. Warehouse design and control: Framework and literature review. *European Journal of Operational Research*, 122(3):515–533, May 2000. Citado na página 52.
- [88] Venkata Reddy Muppani (Muppant) and Gajendra Kumar Adil. A branch and bound algorithm for class based storage location assignment. *European Journal of Operational Research*, 189(2):492–507, September 2008. Citado na página 52.
- [89] V. Muppanimupptant and G. Avil. Efficient formation of storage classes for warehouse storage location assignment: A simulated annealing approach<sup>73</sup>. *Omega*, 36(4):609–618, August 2008. Citado na página 52.

- [90] David Füßler, Nils Boysen, and Konrad Stephan. Trolley line picking: storage assignment and order sequencing to increase picking performance. *OR Spectrum*, 41(4):1087–1121, November 2019. Citado na página 52.
- [91] H. Donald Ratliff and Arnon S. Rosenthal. Order-picking in a rectangular warehouse: A solvable case of the traveling salesman problem. *Operations Research*, 31(3):507–521, June 1983. Citado na página 52.
- [92] Alok Singh and Anurag Singh Baghel. A new grouping genetic algorithm for the quadratic multiple knapsack problem. In *Evolutionary Computation in Combinatorial Optimization*, pages 210–218. Springer Berlin Heidelberg, 2007. Citado na página 54.
- [93] Shyam Sundar and Alok Singh. A swarm intelligence approach to the quadratic multiple knapsack problem. In *Neural Information Processing. Theory and Algorithms*, pages 626–633. Springer Berlin Heidelberg, 2010. Citado na página 54.
- [94] Sang-Moon Soak and Sang-Wook Lee. A memetic algorithm for the quadratic multiple container packing problem. *Applied Intelligence*, 36(1):119–135, August 2010. Citado na página 54.
- [95] David Bergman. An exact algorithm for the quadratic multiknapsack problem with an application to event seating. *INFORMS Journal on Computing*, 31(3):477–492, July 2019. Citado na página 55.
- [96] Martin W. P. Savelsbergh. Branch and price: Integer programming with column generation. In *Encyclopedia of Optimization*, pages 218–221. Springer US, 2001. Citado na página 55.
- [97] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499. Santiago, Chile, 1994. Citado na página 60.
- [98] Sotiris Kotsiantis and Dimitris Kanellopoulos. Association rules mining: A recent overview. *GESTS International Transactions on Computer Science and Engineering*, 32(1):71–82, 2006. Citado na página 60.
- [99] D-Wave: When Was D-Wave Founded? [https://docs.dwavesys.com/docs/latest/c\\_qpu\\_timing.html](https://docs.dwavesys.com/docs/latest/c_qpu_timing.html). Acesso em 30/09/2023. Citado na página 75.
- [100] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014. Citado na página 79.
- [101] Página Inicial: AQT. <https://www.aqt.eu/>. Acesso em 30/09/2023. Citado na página 79.

- [102] White Paper - Hybrid Solver for Constrained Quadratic Models. [dwavesys.com/media/vxunqjit/14-1055a-a\\_hybrid\\_solver\\_for\\_constrained\\_quadratic\\_models.pdf](https://dwavesys.com/media/vxunqjit/14-1055a-a_hybrid_solver_for_constrained_quadratic_models.pdf). Acesso em 30/09/2023. Citado na página 79.