

Joint Optimal Design for Speed and Routing in Maritime Logistics for Green Supply Chain: A Quantum Approximate Optimization Algorithm Approach

Vu Phong Pham, *Student Member, IEEE*, Dang Van Huynh, *Member, IEEE*, Elif Ak, *Member, IEEE*, Long D. Nguyen, *Member, IEEE*, Berk Canberk, *Senior Member, IEEE*, Octavia A. Dobre, *Fellow, IEEE*, Trung Q. Duong, *Fellow, IEEE*

Abstract—Maritime transportation is essential for global trade but presents significant environmental challenges due to its greenhouse gas emissions. Existing studies have addressed these challenges through integrated routing and speed optimization frameworks, yet frequently lack explicit quantification of environmental impacts and exhibit limited scalability for large-scale ship routing operations. Conversely, existing quantum optimization research in vehicle routing predominantly targets land-based transportation scenarios, restricting its direct applicability to maritime logistics. Maritime logistics inherently involve distinct operational complexities, such as nonlinear interactions among speed, payload, fuel consumption, and numerous operational uncertainties. These combined limitations underscore the critical need for quantum optimization methods explicitly designed for green maritime supply chains. To bridge this gap, this paper proposes an efficient quantum-centric optimization framework that uses the quantum approximate optimization algorithm (QAOA) to jointly optimize ship routing and speed management within sustainable maritime supply chains. Specifically, we formulate an NP-hard cost minimization problem integrating critical maritime parameters, including fuel consumption, payload constraints, and operational speeds. We further develop a hybrid quantum-classical alternating optimization approach that iteratively addresses routing decisions through quantum computing techniques and optimizes ship speed using an analytical solution. Simulation results and real quantum hardware experiments demonstrate that our quantum-centric methodology achieves substantial cost reductions and highlights the potential for practical applicability in realistic maritime operations, significantly outperforming

classical optimization benchmarks.

Index Terms—Quantum computing, quantum approximate optimization algorithm, ship routing, green supply chain, route and speed optimization

I. INTRODUCTION

Maritime transportation, carrying over 80% of global trade by volume, is crucial for the international economy; however, its greenhouse gas emissions constitute approximately 3% of global totals and have grown by 20% over the past decade, potentially rising to 130% of 2008 levels by 2050 if unchecked [1]. Its vital role in the global economy underscores the importance of optimizing maritime logistics, which traditionally faces significant challenges due to volatile fuel prices, uncertainties in cargo demands, and increasingly stringent environmental regulations [2]. The shipping industry has therefore become a critical domain for both industrial and academic research, driven by the necessity to balance multiple, often conflicting, demands such as operational efficiency, cost minimization, ecological sustainability, and improved service quality. Consequently, the urgency of incorporating environmentally sustainable practices within maritime logistics has intensified significantly. Air pollution from ships, mainly CO₂, SO_x, and NO_x emissions, represents a major environmental concern, contributing substantially to global warming, climate change, acid rain, and adverse human health effects, particularly in coastal and port areas [1]. This has attracted increased scrutiny from policymakers, resulting in the introduction of rigorous environmental regulations, such as emission control areas (ECAs), carbon taxation schemes, and stringent fuel sulfur limits [3]. Consequently, maritime stakeholders face a pressing obligation to adopt green supply chain practices to comply with these evolving regulatory frameworks and societal expectations.

Moreover, fuel consumption constitutes a significant portion (often exceeding 50%) of total maritime operational costs, directly linking economic viability with ecological sustainability [4]. Inefficient routing and speed practices exacerbate fuel wastage, leading not only to higher operational costs but also to increased environmental pollution [5]. Hence, optimizing ship routing and speed management has become a key strategy to simultaneously reduce operational expenses and mitigate the environmental impact of shipping activities.

Despite considerable research efforts, the complexities inherent in maritime logistics optimization, such as multi-

V. P. Pham, D. V. Huynh, E. Ak and O. A. Dobre are with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1B 3X5, Canada (e-mail: {vppham, vdhuynh, elif.ak, odobre}@mun.ca).

B.Canberk is with the School of Engineering and Built Environment, Edinburgh Napier University, Edinburgh EH10 5DT, UK, (e-mail: b.canberk@napier.ac.uk).

L. D. Nguyen is with Duy Tan University, Da Nang 550000, Vietnam (email: nguyendinhlong1@duytan.edu.vn).

T. Q. Duong is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada and also with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, U.K., (e-mail: tduong@mun.ca).

This paper has been accepted in part for presentation at International Conference on Quantum Communications, Networking, and Computing (QCNC 2025), Nara, Japan, March 2025.

This work was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant Program RGPIN-2025-04941, in part by the Ocean Frontier Institute (OFI) Seed Funding Program, and in part through the Canada First Research Excellence Fund (CFREF) for the project Transforming Climate Action (TCA). The work of B. Canberk is supported in part by The Scientific and Technological Research Council of Turkey (TUBITAK) Frontier R&D Laboratories Support Program for BTS Advanced AI Hub: BTS Autonomous Networks and Data Innovation Lab Project 5239903. The work of O. A. Dobre was supported in part by the Canada Research Chair Program CRC-2022-00187.

dimensional nonlinear relationships between speed, payload, and fuel consumption, pose formidable challenges. Traditional optimization models typically involve complex mixed-integer programming formulations, further complicated by unpredictable operational disturbances such as port congestion, draft restrictions, tidal conditions, and weather uncertainties [6], [7]. Therefore, there is an ongoing need for advanced, robust, and computationally efficient optimization methodologies capable of addressing these multifaceted constraints in real-world maritime scenarios.

Recently, quantum optimization has emerged as a promising solution for dealing with intractable combinatorial optimization problems. The advancement of quantum hardware and hybrid quantum-classical optimization methods presents significant potential for addressing medium- to large-scale optimization problems [8]. Researchers in both industry and academia are actively developing effective approaches to enhance the practicality and universal applicability of quantum solutions for real-world problems. Several advanced quantum algorithms have been introduced [9], [10], such as the quantum approximate optimization algorithm (QAOA), quantum annealing, and the variational quantum eigensolver (VQE). These approaches have demonstrated their potential in solving optimization problems through impressive experimental results obtained from both simulators and real quantum hardware processors. In this context, we propose an optimal design for a joint speed and routing problem (JSRP) in shipping. Specifically, we formulate a cost-minimization problem that considers various factors such as fuel consumption, payload, and ship speed. To address the challenging formulated NP-hard problem, we propose an alternating optimization (AO) approach that iteratively optimizes routing decision variables using a quantum-centric optimization method while handling speed variables with a closed-form solution.

A. Literature Review

In transportation systems, researchers have attempted to employ quantum optimization as low computational complexity and inexpensive for investigations [11]. Among the various challenges in supply chain logistics, the vehicle routing problem (VRP) stands as one of the most classical problems and is widely studied. VRP usually involves routing selection, vehicle determination, and quality-of-service conditions with different objectives such as total cost minimization [12], [13] or total distance minimization [14]. For example, a study tackled multi-objective vehicle scheduling for shipyard operations, achieving notable fuel and emissions reductions [15]. Similarly, another study provided robust formulations for vehicle routing problems in green supply chains and third-party logistics, optimizing costs, emissions, and vehicle utilization efficiency [16]. This intricate nature of VRP makes it an ideal problem for quantum algorithmic approaches, which can potentially offer superior solutions compared to classical methods. In particular, in [12], a multi-truck VRP for supply chain logistics is investigated the total cost optimization in an enormous commercial world by constructing small quadratic unconstrained binary optimizations (QUBO) instances for one-truck VRPs and employing quantum annealing algorithms

to solve these subproblems iteratively. Generally, quantum algorithms can be utilized in solving individual truck routing problems, which are usually intractable and challenging to solve. Quantum methods can be combined with other optimization methods to find the optimal solution, e.g. [13] explores the use of combinatorial optimization (CO) and VQE in a basic VRP solver and analyzes the effect of noise models on VRP circuits. However, depending on different choices of QUBO solvers, the performance characteristics of the proposed algorithms may vary. This result is concluded in [14], where a small VRP with time windows is considered. In particular, the results reveal that QAOA performs better with a sufficient number of iterations, and VQE performs better when the samples are limited. Despite the significant advance and promising outcomes demonstrated by quantum optimization approaches in addressing various VRP models, these studies predominantly focus on land-based transportation scenarios. Consequently, their direct applicability to maritime operations remains limited, as maritime logistics entail distinct characteristics, constraints, and complexities that differ substantially from conventional land-based VRPs.

In the scope of ship routine optimization, recent literature has emphasized the critical importance of multi-objective optimization in green maritime routing. A prominent study developed a joint optimization strategy that integrates ship scheduling with speed adjustments to minimize port waiting times and reduce associated emissions, effectively demonstrating practical emission reductions [17]. Another research also introduced a routing and speed optimization framework that carefully balances operational costs with emissions, particularly within ECAs [18]. A further significant effort presented a comprehensive routing and speed optimization model that explicitly accounts for payload-dependent fuel consumption, providing realistic insights into emission dynamics and operational efficiency [19]. Another study integrated sustainable ship routing, bunkering strategies, and recovery from operational disruptions, providing a holistic view of sustainable maritime logistics [20]. Another approach conceptually formalized the Green Ship Routing and Scheduling Problem (GSRSP), introducing various mathematical formulations integrating emissions directly into routing and scheduling models [21]. Despite these significant contributions, a common limitation across this research cluster is the assumption of deterministic scenarios, with relatively little accommodation for uncertainty such as changing weather conditions, disruptions, or dynamic port congestion. Moreover, these optimization problems, typically formulated as complex nonlinear mixed-integer programs, face significant computational challenges limiting scalability for real-world scenarios.

Innovative quantum and AI-based approaches have emerged as potential solutions to handle the complexity and scalability issues in maritime logistics optimization. The study employed a quantum genetic algorithm enhancing ship speed optimization under multiple regulatory constraints, demonstrating improvements in optimization efficiency [5]. Additionally, quantum-inspired algorithms have been applied effectively to scenarios such as collision avoidance and dynamic multi-objective routing optimization, potentially yielding significant

indirect environmental advantages through reductions in unnecessary vessel maneuvers and associated fuel consumption [22], [23]. Further studies introduced quantum annealing methodologies specifically tailored for complex delivery routing and shipment rerouting problems, demonstrating substantial scalability improvements and efficiency when addressing large-scale optimization tasks [24], [25]. Collectively, these works highlight the capability of quantum computing techniques (such as quantum annealing, QAOA, VQE) to solve complex ship routing problems efficiently. They offer promising directions toward hybrid quantum-classical algorithms that could be scaled effectively to real-world operational scenarios. However, explicit modeling and quantification of environmental benefits such as emissions reductions are often missing, leaving room for improvement in directly connecting these advanced computational methods to sustainability metrics.

The integration of advanced communication technologies and digital infrastructures into maritime operations represents another vital research trend for sustainable maritime management. Recent research introduced innovative maritime network architectures leveraging AI-driven edge intelligence and metaverse technologies, enabling improved energy efficiency and more responsive real-time operational decision-making [26]. Another notable study developed a robust, multi-objective weather routing algorithm based on hybrid particle swarm optimization, significantly reducing both fuel consumption and voyage risks under diverse weather conditions [27]. Additionally, deep learning-based approaches utilizing Space-Air-Ground Integrated Networks (SAGIN) have been proposed, demonstrating enhanced maritime communication reliability and coverage. These advancements indirectly contribute to operational sustainability by ensuring consistent connectivity and reducing vessel idle times [28]. Despite these valuable advancements, current research in this domain frequently overlooks explicit modeling of emissions caused by insufficient ship routing, often relying instead on indirect inferences about environmental impacts.

B. Motivation and Contributions

Recent studies have highlighted the importance of multi-objective strategies for sustainable ship routing and speed optimization. These strategies involve joint scheduling and speed reduction to minimize emissions at ports [17] and payload-dependent optimization frameworks designed to balance emissions with operational costs [19]. Moreover, innovative quantum and quantum-inspired optimization methods have emerged as promising alternatives to classical techniques, exhibiting superior computational efficiency and scalability potential.

Motivated by these developments and the existing research gaps, this paper aims to bridge these gaps with the following contributions:

- We formulate a comprehensive optimization problem for ship routing and speed, explicitly accounting for fuel consumption, payload constraints, and operational speed to minimize overall maritime operational costs.
- We develop an efficient alternating optimization (AO) framework that employs a QAOA for routing decisions

combined with a closed-form analytical solution for optimal speed determination.

- Complementing the AO-based solution, we propose a QUBO-based solution that jointly solves optimal speed and routing decisions within a quantum-centric optimization framework.
- We validate the effectiveness and robustness of the proposed quantum-centric optimization framework using extensive numerical simulations. The results demonstrate significant cost reductions compared to classical benchmark algorithms and show optimal solutions equivalent to brute-force methods.
- We demonstrate the practical applicability and scalability of quantum optimization methods to realistic maritime logistics scenarios, highlighting their suitability for addressing complex optimization challenges within green supply chain management.

C. Paper Structure and Notations

The remainder of the paper is structured as follows. Section II introduces the system model and formulates the joint optimization problem for ship routing and speed. Section III provides fundamentals of quantum optimization, including key principles of quantum computing, quadratic unconstrained binary optimization (QUBO), and the quantum approximate optimization algorithm (QAOA). Section IV details the proposed alternating optimization (AO) framework for solving the routing and speed optimization problem. Section V develops a joint optimization solution for speed and routing decisions based on the QUBO transformation. Section VI presents numerical simulation results along with comprehensive discussions. Finally, Section VII concludes the paper by summarizing the main contributions and outlining future research directions.

Notations: Throughout the paper, numbers are represented by lowercase letters, while matrices and vectors are denoted by bold uppercase and lowercase letters, respectively. \mathbb{C} and \mathbb{R} denote the set of complex numbers and real numbers, whereas $\mathbb{R}_{\geq 0}^2$ represents the non-negative real numbers, and the symbol \dagger denotes the Hermitian transpose. The bold letters, for example, \mathbf{x} , \mathbf{v} are used to describe the vectors of variables, while regular letters such as x_{ki} or n are given parameters or a specific variable in the vectors of variables.

II. SHIPMENT ROUTING MODEL AND PROBLEM FORMULATION

We consider a shipment routing system, as illustrated in Fig. 1. The routing system is regarded to have n ships and each of which is assigned to pick up n cargoes from their respective harbors. Once these cargoes are collected, they must be transported to their designated consumption ports. Initially, ships are located in their *depots*, or central ports, and are required to return to their depots before the limited operation time T_{\max} . For simplicity, we can define the loading ports, discharging ports and ship depots as nodes.

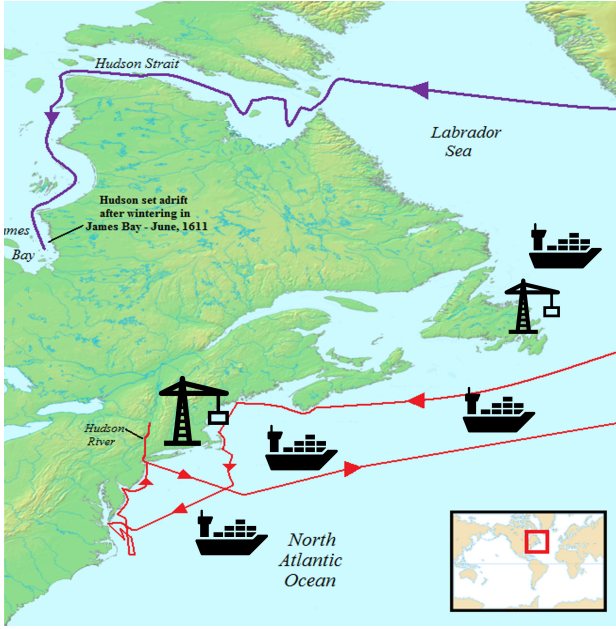


Fig. 1: An illustration of the vehicle routing system model, depicting the transportation of cargoes between multiple harbors by ships.

A. Shipment Routing Model

Each cargo has its weight w_i and can be associated with its port, denoted as i , while the desired destination of the cargo i can be represented by $i + n$. Let $\mathcal{L} = \{1, 2, \dots, n\}$ and $\mathcal{U} = \{n + 1, n + 2, \dots, 2n\}$ be the set of loading and discharging nodes, respectively. The ships can be denoted as k , and we indicate $\mathcal{D} = \{d(1), d(2), \dots, d(n)\}$ as the set of ships depot. Each ship has its original weight m_k , which contains people on the ships, oils or consumables, but does not have any cargoes on the board yet. Each ship k must operate at a constant speed $v_k \in [v_{\min}, v_{\max}]$ throughout the transportation process. Let o_i and o_{i+n} be the unchanged pickup and delivery time on the nodes i and $i + n$. The route leg from node A to node B can be defined as (A, B) , and we use $d_{A,B}$ to describe the distance between these two nodes.

According to [29], the realistic fuel consumption f_k of ship k sailing through route leg (A, B) can be approximated by the ship speed v_k as

$$f_{A,B}^k(v_k) = K(p + v_k^q)(w_{A,B}^k + m_k)^{\frac{2}{3}}, \quad (1)$$

where $w_{A,B}^k$ is the payload of ship k in leg (A, B) and $K > 0$, $p \geq 0$, $q \geq 3$ are known ship-related constants. Furthermore, if the payload $w_{A,B}^k$ is fixed, then the total cost of the ship k traveling through the leg (A, B) can be expressed as

$$C_{A,B}^k(v_k) = [P_{\text{fuel}} f_{A,B}^k(v_k) + \alpha u + \beta w_{A,B}^k + F] \frac{d_{A,B}}{v_k}, \quad (2)$$

where P_{fuel} is the fuel purchased at a known price (\$/tons), α is the unit cargo port inventory cost (\$/tons/day), β is the unit cargo in-transit inventory cost (\$/tons/day), u is the weight of cargoes not loaded yet, and F is the known freight rate (\$/day).

Let x_{ki} be the binary decision variable, where $x_{ki} = 1$ if

ship k decides to transport cargo i and $x_{ki} = 0$ otherwise. During the entire operation time T_{\max} , each cargo must be transferred by exactly one ship. This requirement can be mathematically presented as

$$\begin{aligned} \sum_{i \in \mathcal{L}} x_{ki} &= 1, \forall k \in \mathcal{K}, \\ \sum_{k \in \mathcal{K}} x_{ki} &= 1, \forall i \in \mathcal{L}. \end{aligned} \quad (3)$$

Herein, the first equation ensures that for all cargoes that are available, each ship can deliver only one cargo to its associated port. The second formulation implies that each cargo can only be chosen once. This equation also helps to avoid a special case where a cargo is picked up by many ships, which is unrealistic in maritime transportation. In this studied routing network, speed plays a pivotal role in further optimizing the problem. The speed range can be written as

$$v_{\min} \leq v_k \leq v_{\max}, \forall k \in \mathcal{K}, \quad (4)$$

where v_{\min} and v_{\max} are the lowest and highest speed that ships are allowed to operate in this model. It can be observed that the transportation time of each ship does not exceed the maximum allowable operation time, T_{\max} . Hence, we have the following constraint described as follows:

$$\begin{aligned} \sum_{i \in \mathcal{L}} x_{ki} \left(\frac{d_{d(k),i}}{v_k} + o_i + \frac{d_{i,i+n}}{v_k} + o_{i+n} + \frac{d_{i+n,d(k)}}{v_k} \right) \\ \triangleq T_k(\mathbf{x}_k, v_k) \leq T_{\max}, \quad \forall k \in \mathcal{K}, i \in \mathcal{L}, \end{aligned} \quad (5)$$

where $T_k(\mathbf{x}_k, v_k)$ denoted as the total time ship k travels with unchanged speed v_k and $\mathbf{x}_k \triangleq \{x_{ki}\}_{i \in \mathcal{L}}$ be the set of binary decision variables around ship k . The speed function $T_k(\mathbf{x}_k, v_k)$ is formulated by the combination of three stages of transport. The first stage is port call, in which the ships are headed to the loading nodes to receive the cargo. In the next stage, ships transport the cargo to their designated ports of discharge. The final stage is when ships return to their depots after the cargo has been returned to its corresponding harbor.

B. Optimization Problem Formulation

Let $\mathbf{x} \triangleq \{\mathbf{x}_k\}, k \in \mathcal{K}$ and $\mathbf{v} \triangleq \{v_k\}, k \in \mathcal{K}$ be the set of binary decision variables and the ships speed. The total cost for using ship k throughout the transport process can be formulated as

$$\begin{aligned} C_k(\mathbf{x}_k, v_k) \\ \triangleq \sum_{i \in \mathcal{L}} x_{ki} \left(C_{d(k),i}^k(v_k) + C_{i,i+n}^k(v_k) + C_{i+n,d(k)}^k(v_k) \right) \\ = \sum_{i \in \mathcal{L}} \frac{x_{ki}}{v_k} \left\{ d_{d(k),i} \left[P_{\text{fuel}} f_{d(k),i}^k(v_k) + \alpha w_i + F \right] \right. \\ \quad + d_{i,i+n} \left[P_{\text{fuel}} f_{i,i+n}^k(v_k) + \beta w_i + F \right] \\ \quad \left. + d_{i+n,d(k)} \left[P_{\text{fuel}} f_{i+n,d(k)}^k(v_k) + F \right] \right\} \\ = \sum_{i \in \mathcal{L}} x_{ki} S_{ki}(v_k), \end{aligned} \quad (6)$$

where $S_{ki}(v_k) \triangleq \frac{1}{v_k} \left\{ d_{d(k),i} \left[P_{\text{fuel}} f_{d(k),i}^k(v_k) + \alpha w_i + F \right] + d_{i,i+n} \left[P_{\text{fuel}} f_{i,i+n}^k(v_k) + \beta w_i + F \right] + d_{i+n,d(k)} \left[P_{\text{fuel}} \times f_{i+n,d(k)}^k(v_k) + F \right] \right\}$. Similar to the time function of the ship, the total cost of each ship is also the combination of the transportation stages. The final objective is to minimize the total routing costs of all ships in the considered network. Based on the formulations of the total cost of the ships along with the system constraints set above, our original optimization problem is mathematically formulated as follows:

$$\min_{\mathbf{x}, \mathbf{v}} \sum_{k \in \mathcal{K}} C_k(\mathbf{x}_k, v_k), \quad (7a)$$

$$\text{s.t.} \sum_{i \in \mathcal{L}} x_{ki} = 1, \quad \forall k \in \mathcal{K}, \quad (7b)$$

$$\sum_{k \in \mathcal{K}} x_{ki} = 1, \quad \forall i \in \mathcal{L}, \quad (7c)$$

$$v_{\min} \leq v_k \leq v_{\max}, \quad \forall k \in \mathcal{K}, \quad (7d)$$

$$T_k(\mathbf{x}_k, v_k) \leq T_{\max}, \quad \forall k \in \mathcal{K}, \quad (7e)$$

$$x_{ki} \in \{0, 1\}, \quad \forall k \in \mathcal{K}, i \in \mathcal{L}. \quad (7f)$$

The objective (7a) minimizes the total cost of the routing network. Constraint (7b) shows that each ship can arrive at exactly one loading node, and (7c) ensures that each loading node serves only one ship. Constraint (7d) describes the upper bounds and lower bounds of the speed of the ships. Constraint (7e) ensures that the operation time of each ship does not exceed the known time limitation T_{\max} . Lastly, constraint (7f) represents the binary variables.

III. PRELIMINARIES ON QUANTUM OPTIMIZATION

Quantum optimization is widely acknowledged for its remarkable performance in solving NP-hard problems, i.e., extremely challenging and time-consuming to find the optimal value computationally. In this section, we address the fundamental concepts of quantum optimization. Additionally, we also introduce the QUBO formulation and the QAOA algorithm, which are among the most crucial in quantum optimization and are the main method used in this study.

A. Quantum Computing Fundamentals

In classical computing, bits are considered the most basic concept as their values can only be '0' or '1'. The combinations of countless bits have structured numerous components of classical computing, including numbers, tasks, logic gates, etc. Likewise, quantum computing has its own 'bits', which are usually called quantum bits, or *qubits*, along with their unique denotations as $|0\rangle$ and $|1\rangle$. However, qubits are not limited by the trivial values '0' and '1' as their fellows, but they are demonstrated as conjunctions of ' $|0\rangle$ ' and ' $|1\rangle$ '. Generally, the formulation of a qubit can be represented as

$$|\psi\rangle = a|0\rangle + b|1\rangle, \quad (8)$$

where $a, b \in \mathbb{C}$ and $|a|^2 + |b|^2 = 1$. This state can also be referred to as *superposition*, which contains $|0\rangle, |1\rangle$ and

their integration. The parameters a and b are usually called *probability amplitudes*. This qubit state is often referred to as *ket*, and we also have the dual of ket defined as *bra*. Specifically, the denotation of bra can be described as

$$\langle\psi| = |\psi\rangle^\dagger.$$

Although the values of a and b can be addressed by some theoretical calculations, the outputs of the qubits are only $|0\rangle$ or $|1\rangle$. This implies that the precise values of complex numbers a and b cannot be directly observed in practice. However, there are some alternative tools that can be used to estimate these coefficients.

The process of extracting information about qubits in $|0\rangle$ and $|1\rangle$ is called *measurement*, and we say that the quantum system *collapses* into one of these two states. The probabilities for a qubit to collapse into $|0\rangle$ and $|1\rangle$ are $|a|^2$ and $|b|^2$. These concepts are crucial for quantum computing, as the execution processes of qubits are sometimes sophisticated and computationally expensive to analyze. By operating these quantum systems with a sufficiently large number of iterations, we can approximate the probabilities that these qubits collapse to $|0\rangle$ or $|1\rangle$.

Now, we apply the Euler formula on the complex components a and b to have

$$a = r_0 e^{i\varphi_0}, b = r_1 e^{i\varphi_1},$$

where $r_0, r_1 \in \mathbb{R}_{\geq 0}^2$ are *amplitudes* and $\varphi_0, \varphi_1 \in [0, 2\pi]$ are *phases* of a and b , respectively. In this way, $|\psi\rangle = r_0 e^{i\varphi_0} |0\rangle + r_1 e^{i\varphi_1} |1\rangle$ has a probability of r_0^2 that collapses into $|0\rangle$, and a probability of r_1^2 that collapses into $|1\rangle$. We can rewrite $|\psi\rangle$ as

$$|\psi\rangle = e^{i\varphi_0} \left(r_0 |0\rangle + r_1 e^{i(\varphi_1 - \varphi_0)} |1\rangle \right) = e^{i\varphi_0} |\phi\rangle. \quad (9)$$

Notice that the possibilities of $|\phi\rangle$ collapsed into $|0\rangle$ and $|1\rangle$ are equivalent to $|\psi\rangle$ as r_0^2 and r_1^2 . Hence, these two superpositions cannot be distinguished by measurements. The scalar $e^{i\varphi_0}$ is called a *global phase* and is not *observable*.

Since qubits are expressed of general qubits as $|0\rangle, |1\rangle$, it is important to understand their geometric representation. The equation (9) showed that the phase difference $\varphi_1 - \varphi_0$ has a great impact on superpositions rather than φ_0 or φ_1 itself. Furthermore, the magnitudes r_0 and r_1 are constrained by $r_0^2 + r_1^2 = 1$, which is ideal for trigonometric parameterization. Therefore, by denoting $\varphi \triangleq \varphi_1 - \varphi_0$ and $r_0 = \cos(\theta/2), r_1 = \sin(\theta/2)$ where $\theta \in [0, \pi]$, the superposition can be reformulated as

$$|\phi\rangle = \cos \frac{\theta}{2} |0\rangle + \sin \frac{\theta}{2} e^{i\varphi} |1\rangle, \quad (10)$$

which is identical to the spherical coordinates system (ρ, θ, φ) . To formulate our spherical qubit representation, we need to find the 'poles' of the axes.

For z-axis poles: In the spherical coordinate system, the z-axis north pole satisfies $\varphi = 0$ and $\theta = 0$ and the south pole is the point complies with $\varphi = 0$ and $\theta = \pi$. Substituting this into (10), we have

$$|\phi\rangle = \cos 0 |0\rangle + \sin 0 |1\rangle = |0\rangle,$$

and

$$|\phi\rangle = \cos \frac{\pi}{2} |0\rangle + \sin \frac{\pi}{2} |1\rangle = |1\rangle.$$

For x-axis poles: The x-axis poles are the points that satisfy $\theta = \pi/2$ and $\varphi = 0$ as the right pole or $\varphi = \pi$ as the left pole. Substituting this into (10), we have

$$|\phi\rangle = \cos \frac{\pi}{4} |0\rangle + \sin \frac{\pi}{4} |1\rangle = \frac{\sqrt{2}}{2} (|0\rangle + |1\rangle),$$

and

$$|\phi\rangle = \cos \frac{\pi}{4} |0\rangle - \sin \frac{\pi}{4} |1\rangle = \frac{\sqrt{2}}{2} (|0\rangle - |1\rangle).$$

Usually, the denotations $|+\rangle$ and $|-\rangle$ are used instead of $\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$ and $\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$, known as the *plus state* and the *minus state*.

For y-axis poles: The poles on the y-axis are the points that satisfy $\theta = \pi/2$ and $\varphi = \pi/2$ as the north pole or $\varphi = 3\pi/2$ as the south pole. Substituting this into (10), we have

$$|\phi\rangle = \cos \frac{\pi}{4} |0\rangle + \sin \frac{\pi}{4} |1\rangle e^{i\frac{\pi}{2}} = \frac{\sqrt{2}}{2} (|0\rangle + i|1\rangle),$$

and

$$|\phi\rangle = \cos \frac{\pi}{4} |0\rangle + \sin \frac{\pi}{4} |1\rangle e^{i\frac{3\pi}{2}} = \frac{\sqrt{2}}{2} (|0\rangle - i|1\rangle).$$

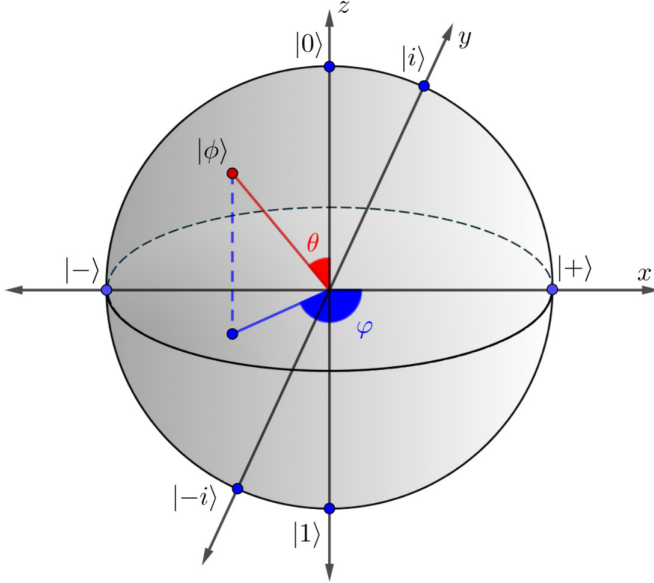


Fig. 2: An illustration of the Bloch sphere in quantum computing with x, y, z-axis poles and the superposition $|\phi\rangle$ performed by the combination of spherical coordinates θ and ϕ .

We can also denote $|i\rangle$ for $\frac{\sqrt{2}}{2} (|0\rangle + i|1\rangle)$ and $|-i\rangle$ for $\frac{\sqrt{2}}{2} (|0\rangle - i|1\rangle)$ [30]. Moreover, every state of a qubit can be represented as points on this unit sphere, and this sphere is called a *Bloch sphere* in quantum computing. As illustrated in Figure 2, the superposition $|\phi\rangle$ is demonstrated by the angles θ, φ that are associated with the x-axis and the z-axis, respectively.

We have seen various combinations of multiple bits in classical computing, and it is even more complicated in quantum

computing. The basic forms of a 2-vector qubit can be written as follows:

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

For simplicity, $|00\rangle, |01\rangle, |10\rangle$ and $|11\rangle$ can be rewritten in a decimal form as $|0\rangle, |1\rangle, |2\rangle$ and $|3\rangle$. The configuration of the 2 qubit superposition is given by

$$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle + a_2 |2\rangle + a_3 |3\rangle, \quad (11)$$

where $a_i \in \mathbb{C}, 0 \leq i \leq 3$, and $\sum_{i=1}^3 |a_i|^2 = 1$. In high multi-dimensional vector space, a basic n qubit form in $M = 2^n$ vectors can be enumerated as follows:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, |M-1\rangle = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}. \quad (12)$$

Equivalently with (11), the superposition of the qubits can be formulated as

$$|\psi\rangle = \sum_{i=0}^{M-1} a_i |i\rangle, \quad (13)$$

where $a_i \in \mathbb{C}, 0 \leq i \leq M-1$ and $\sum |a_i|^2 = 1$.

In quantum mechanics, quantum states are mathematical representations that capture all the information that exists in a quantum system. By changes in time, the information about these states updates itself, and this stage is called *evolution*. This evolution is uniformly described by a time-dependent equation, or specifically, the *Schrödinger equation* [30]. This notable equation can be mathematically expressed as

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = \hat{H} |\psi(t)\rangle, \quad (14)$$

where i is the imaginary unit, \hbar is the reduced Planck constant, and \hat{H} is the *Hamiltonian* formulation that describes the total energy of the system. In general, the solution of this Hamiltonian is given as

$$|\psi(t)\rangle = U(t) |\psi(0)\rangle, \quad (15)$$

where $U(t)$ denoted as an *operator* from state $|0\rangle$ at the time t . The operator $U(t)$ may have different formulations depending on whether the Hamiltonian is time-dependent or not. The time-dependent case is more complicated, but we can derive the result of (14) in the case our Hamiltonian H is time-independent as

$$U(t) = e^{-iHt/\hbar}. \quad (16)$$

From this part, we will assume that $\hbar = 1$. Even if the Hamiltonian formulation depends on time, it is proven that with any fixed time t_0 , $U = U(t_0)$ is always a *linear operator* [31]. This concept can be described as

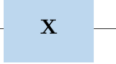

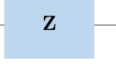
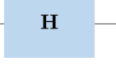
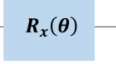
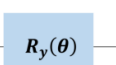
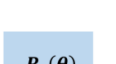
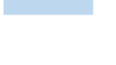
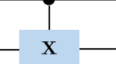

$$U(a|\psi\rangle + b|\phi\rangle) = aU|\psi\rangle + bU|\phi\rangle. \quad (17)$$

Moreover, U is also *unitary* and can be formally written as

$$UU^\dagger = U^\dagger U = 1, \quad (18)$$

where U^\dagger denoted as the matrix attained by the Hermitian transpose of U , or so called, the *adjoint* of U . These characteristics help U to address superpositions - a combination of basic qubits.

TABLE I: Some Basic Quantum Gates with Figure Symbols

Operator	Gate	Matrix
Pauli-X		$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Pauli-Y		$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
Pauli-Z		$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Hadamard		$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
$R_x(\theta)$		$\begin{pmatrix} \cos(\frac{\theta}{2}) & -i \sin(\frac{\theta}{2}) \\ -i \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}$
$R_y(\theta)$		$\begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}$
$R_z(\theta)$		$\begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$
CNOT		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
CY		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{pmatrix}$
CZ		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$

In problem solving, quantum computing employs quantum circuits, which are made up of multiple quantum logic gates to address various types of computational tasks. These quantum logic gates are quite analogous to logic gates in classical computing and can be used to manipulate the operations of qubits. However, with superposition and entanglement, quantum gates have surpassed the limitations of classical logic gates on binary operations. To implement these quantum effects, quantum gates are classified according to how many qubits they influence. In particular, there are two types of quantum gates, named as single and multi qubit gates. While single qubit gates are used to handle the states of one qubit, multi

qubit gates are utilized in executing the states of multiple qubits. The operation stage of quantum gates can be expressed by matrices of size $2^n \times 2^n$, where n is the number of qubits on which the quantum gates have to settle. Table I illustrates some important single and double qubit gates along with the representative matrices. Since superpositions are combinations of basic qubits, the effect of quantum gates on superpositions can be analyzed by how they operate on the qubits.

1) Single qubit gates

In Bloch sphere, single qubit gates are translations of any angle around any axis. Here are some of the essential single qubit gates, which act as the angle rotations around one axis.

Pauli-X gate (X): The X-gate allows the poles of the x-axis to rotate their positions. This gate is also known as the NOT gate. The impact of this gate on other basis states such as $|0\rangle$ and $|1\rangle$ can be uniformly expressed as follows:

$$X|0\rangle = |1\rangle, X|1\rangle = |0\rangle.$$

Pauli-Y gate (Y): In the same way, the Y-gate allows the poles of the y-axis to switch their places. The effect of Y-gate on basic states can be expressed as

$$Y|0\rangle = i|1\rangle, Y|1\rangle = -i|0\rangle.$$

Pauli-Z gate (Z): The Z-gate also allows the poles of the z-axis to interchange their locations. The effect of Z-gate on basic states can be formulated as

$$Z|0\rangle = |0\rangle, Z|1\rangle = -|1\rangle.$$

Hadamard gate (H): The Hadamard gate is one of the most important gates as it can rotate the states between the x-axis and the z-axis. The action of the H-gate can be written as

$$H|0\rangle = |+\rangle, H|+\rangle = |0\rangle,$$

and

$$H|1\rangle = |-\rangle, H|-\rangle = |1\rangle.$$

X-rotation gate ($R_x(\theta)$): If the X-gate exchanges the positions of the poles by a π rotation, $R_x(\theta)$ allows these poles in the x-axis to move at a different angle $\theta \in [0, \pi]$. After being executed, the superposition of the qubits are given as

$$R_x(\theta)|0\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle - i \sin\left(\frac{\theta}{2}\right)|1\rangle,$$

$$R_x(\theta)|1\rangle = -i \sin\left(\frac{\theta}{2}\right)|0\rangle + \cos\left(\frac{\theta}{2}\right)|1\rangle.$$

Y-rotation gate ($R_y(\theta)$): Equivalently, $R_y(\theta)$ allows the poles to move in y-axis by a different angle $\theta \in [0, \pi]$. After being executed, the superposition of the qubits are calculated as

$$R_y(\theta)|0\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle - \sin\left(\frac{\theta}{2}\right)|1\rangle,$$

$$R_y(\theta)|1\rangle = \sin\left(\frac{\theta}{2}\right)|0\rangle + \cos\left(\frac{\theta}{2}\right)|1\rangle.$$

Z-rotation gate ($R_z(\theta)$): $R_z(\theta)$ also permits the poles to move in z-axis by a different angle $\theta \in [0, \pi]$. After being

executed, the superposition of the qubits are expressed as

$$\begin{aligned} R_z(\theta) |0\rangle &= e^{-i\theta} |0\rangle, \\ R_z(\theta) |1\rangle &= e^{i\theta} |1\rangle. \end{aligned}$$

It is also essential to know that we can generate any single qubit gate by combinations of Z-rotation and Y-rotation gates. According to [32], for any single qubit gate U , there exists three angles α, β, γ that satisfies the following equation:

$$U = R_z(\alpha)R_y(\beta)R_z(\gamma). \quad (19)$$

2) multi qubit gates

The operations of the multi qubit systems are also unitary and linear according to the solution of the Schrödinger equation (15). There are many types of multi qubit gates: They can be the product of several one qubit gates that operate independently, or they can arrange certain qubits to control the state of others, which is known as *entanglement* - one of the most exquisite and unique features of quantum computing. In specific, before the measurement stage, there are some qubits chosen as *control qubits* are connected to others. The states of these other qubits will be decided in the measurement stage, in which the control qubits collapse to basic states. Below are some of the multi qubit gates constructed based on entanglement.

Controlled NOT gate (CX): As Table I depicts, the control qubit is indicated by a black dot, and the other is indicated by the X-gate, or sometimes the \oplus symbol. If the control qubit is measured as 0, the other qubit will have the same result as its initial input. However, if the control qubit is measured as 1, this indicates that the X-gate has been activated, and the result of the other qubit will be reversed. The action of this two qubit gate can be described as

$$\begin{aligned} CNOT |00\rangle &= |00\rangle, CNOT |01\rangle = |01\rangle, \\ CNOT |10\rangle &= |11\rangle, CNOT |11\rangle = |10\rangle. \end{aligned}$$

Controlled Y gate (CY): The idea of this gate is equivalent to CX gate, where the control qubit decides whether the other qubit will be manipulated by the Y-gate or not. The action of this two qubit gate can be described as

$$\begin{aligned} CY |00\rangle &= |00\rangle, CY |01\rangle = |01\rangle, \\ CY |10\rangle &= i |11\rangle, CY |11\rangle = -i |10\rangle. \end{aligned}$$

Controlled Z gate (CZ): The Controlled Z gate also execute the qubit according to the state of the control qubit after the measurement stage. The action of this two qubit gate can be described as

$$\begin{aligned} CZ |00\rangle &= |00\rangle, CZ |01\rangle = |01\rangle, \\ CZ |10\rangle &= |10\rangle, CZ |11\rangle = -|11\rangle. \end{aligned}$$

There are many more multi qubit gates that we have not introduced yet. In fact, quantum computing itself cannot physically construct every possible quantum gate, as the exponential growth over the vector space is considered. However, they can still be universally implemented by other gates. As indicated in (19), every single qubit gate can be built based on Z-rotation and Y-rotation gates, and this principle also applies in multi qubit gates. It can be shown that we can implement any multi

qubit gates by using single qubit gates and the CNOT gate. This fact will be crucial in building the *quantum circuit*, which is one of the most practical and popular models in quantum computing.

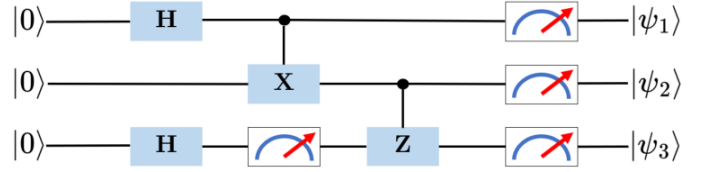


Fig. 3: An example of a quantum circuit processed with 3 qubits.

In classical computing, each computation consists of three main components: input bits, gates, and outputs. Similarly, quantum computing does its computation by input qubits, quantum gates with measurements, and output qubits. As described in Fig. 3, the three input qubits are all $|0\rangle$ and are executed by H-gates, CX, CZ, and measurements, resulting in the eventual qubits as $|\psi_1\rangle, |\psi_2\rangle, |\psi_3\rangle$. The process initializes horizontally from the left side to the right side by *wires*, and the last step before showing the output qubits should be the measurement stage at the end of the process. By quantum circuits, quantum computers and libraries can virtually illustrate the process of computations and utilize quantum algorithms such as VQE, QAOA to minimize any given Hamiltonian.

B. An Introduction to QUBO

QUBO framework represents optimization problems with binary variables in unconstrained quadratic formulations. In specific, we can write down the the general form of a QUBO problem as follows:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \mathbf{x}^T \mathbf{Q} \mathbf{x}, \\ \text{s.t. } & x_i \in \{0, 1\}, i = 1, \dots, n, \end{aligned} \quad (20)$$

where x_i denoted as the i -th component of \mathbf{x} , \mathbf{Q} is a symmetric matrix that describes the linear terms of the variables in \mathbf{x} and the interactions between two different decision variables in \mathbf{x} . The objective of QUBO problems is usually called *cost function*.

QUBO method can be used to mathematically demonstrate numerous real-world problems, as it basically represents 0–1 problems, including graph coloring, scheduling, networking, clustering, etc. Hence, QUBO problems are considered as NP-hard due to their exponential growth in computational complexity. QUBO problems are also very flexible because they can be converted to other optimization problems, such as the Ising model - a special case of Hamiltonian [32], as follows:

$$\begin{aligned} \min_{\mathbf{z} \in \mathbb{R}^n} \quad & - \sum_{i,j} J_{ij} z_i z_j - \sum_i J_i z_i, \\ \text{s.t. } & z_i \in \{-1, 1\}, i = 1, \dots, n. \end{aligned} \quad (21)$$

Herein, J_{ij} are coefficients corresponding to elements and J_i represents the influence of external fields on the particle i .

The variables z_i are also known as *spin variables*, as their states can change by a spin inversion. These spin variables can be represented by Z-gates and thus tackled by quantum optimization algorithms. In particular, if we denote $H_{\text{state}} = -\sum_{i,j} J_{ij} Z_i Z_j - \sum_i J_i Z_i$ and $|\phi\rangle$ as the basis state of the system, the classical minimization can now be converted to find a basis state $|\phi\rangle$ so that the formulation $\langle\phi|H_{\text{state}}|\phi\rangle$ attains its minimum value.

Although the QUBO methodology is tailored for unconstrained binary problems, it is viable to reformulate equality-constrained problems into a QUBO framework by adding suitable penalty terms to the considered objective function. These penalty terms must be sufficiently strong to punish the objective so that the constraints are not invalidated. For instance, the general form of an equality constrained quadratic binary problem can be rewritten as

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}), \\ \text{s.t. } A\mathbf{x} = b, x_i \in \{0, 1\}, i = 1, \dots, n. \end{aligned} \quad (22)$$

where m is the number of equality constraints, $A \in \mathbb{R}^{m \times n}$ is the constraint matrix and $b \in \mathbb{R}^{m \times 1}$ is the constrained vector. We can reformulate this problem as follows:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) + \lambda(A\mathbf{x} - b)^2, \\ \text{s.t. } \mathbf{x} \in \{0, 1\}. \end{aligned} \quad (23)$$

The penalty parameters $\lambda \in \mathbb{R}^{m \times 1}$ are now introduced in the regarded objective. These penalty terms must be chosen wisely, as immense penalties might cause some numeric errors due to the narrow gaps between the coefficients, and slow convergence of the unconstrained objective. In contrast, taking negligible penalties might result in the existence of infeasible solutions corresponding with small values, which may sometimes deceive quantum solvers.

It is also essential that the inequality constraints of binary problems can be converted to equality constraints by the Simplex methodology. The main idea is to add some binary *slack variables* into these constraints to help the other variables satisfy the equality. For example, an inequality constraint $x_1 + 2x_2 \leq 5$ can be rewritten as

$$x_1 + 2x_2 + s_1 + 2s_2 + 2s_3 = 5.$$

In this way, even if x_1 and x_2 are set to 0, the equality is still attained for $s_1 = s_2 = s_3 = 1$. Hence, every quadratic binary problem with inequality terms can be modified to an equality constrained quadratic binary problem, and a QUBO formulation.

C. Quantum Approximate Optimization Algorithm (QAOA)

Before we discuss QAOA in more detail, let us first introduce the Quantum Adiabatic Theorem, which establishes the foundational theoretical framework.

Considering the Schrödinger equation on (14), in the situation where the Hamiltonian is time-dependent. The *ground state* of a Hamiltonian $H(t)$ is defined as the *eigenstate* corresponding to the lowest *eigenvalue* (i.e., the minimum energy

state) of $H(t)$. Mathematically, assumed that the Hamiltonian $H(t)$ has n eigenstates as $|\psi_i(t)\rangle$ associated with the eigenvalues $E_i(t)$, then we have

$$H(t) |\psi_i(t)\rangle = E_i(t) |\psi_i(t)\rangle, i = 1, \dots, n. \quad (24)$$

The ground state of the Hamiltonian $H(t)$ is the eigenstate $|\psi_k\rangle$ associated with the eigenvalue $E_k(t)$, which satisfies (24) and $E_k(t) \leq E_i(t), \forall i \neq k$.

The Quantum Adiabatic Theorem indicates that if the Hamiltonian of a quantum system slowly varies from the state H_1 to H_2 , then the system will remain in the corresponding instantaneous eigenstate throughout evolution [33]. For simplicity, let us denote $|g_1\rangle$ and $|g_2\rangle$ as the ground states of H_1 and H_2 , respectively. We also assume that it takes a period of time t_F for the Hamiltonian to adiabatically evolve from H_1 to H_2 . If the initial state $|\psi(t)\rangle$ of the system follows the Schrödinger equation (14) and is prepared in the ground state of the initial Hamiltonian H_1 , or simply $|\psi(0)\rangle = |g_1\rangle$, then if the evolution is slow enough, the final state of the system $|\psi(t_F)\rangle$ is also the ground state of the final Hamiltonian, or $|\psi(t_F)\rangle = |g_2\rangle$.

We have exploited the idea of the Quantum Adiabatic Theorem, which is a cornerstone of adiabatic quantum computing and other optimization methods. QAOA is also inspired by the adiabatic quantum evolution, where the Hamiltonian slowly changes over time. The core idea of QAOA is that we can derive the approximate solution of the final ground state by discretizing the considered Hamiltonian into a sequence of time frame. The time-dependent Hamiltonian in each time frame will be replaced by a time-independent Hamiltonian, which can be solved directly as (16). Mathematically, in adiabatic quantum computing, the Hamiltonian is in the form of

$$H(t) = A(t)H_1 + B(t)H_2, \quad (25)$$

where $A(t)$ and $B(t)$ are the interpolation functions over the interval $[0, t_F]$ such that $A(0) = B(t_F) = 1$ and $A(t_F) = B(0) = 0$. These notations determine that $H(0) = H_1$ and $H(t_F) = H_2$ as we expected. Let us define Δ_t as a small amount of time, t_i is the i -th time frame and $N = t_F/\Delta_t$ as the number of *layers* in QAOA. If the Δ_t is sufficiently enough, we have

$$H(t) \approx \sum_{j=1}^N A(t_j)H_1 + B(t_j)H_2. \quad (26)$$

It is clear that the approximation is better if the Δ_t decreases. According to the Quantum Adiabatic Theorem and (16), in the first time frame, the ground state evolves to $|\psi(t_1)\rangle = \exp(-i(A(t_1)H_1 + B(t_1)H_2))|\psi(0)\rangle$. If we continue this process until the final time frame, we have our final ground state estimated as follows:

$$|\psi(t_F)\rangle \approx \prod_{j=N}^1 e^{(-i\Delta_t(A(t_j)H_1 + B(t_j)H_2))} |\psi(0)\rangle. \quad (27)$$

Then, we apply the Lie-Trotter formula [34] given as

$$e^{A+B} \approx e^A e^B, \quad (28)$$

where A, B are quantum operators. The final state (27) can be reformulated as

$$|\psi(t_F)\rangle \approx \prod_{j=N}^1 (e^{-i\gamma_j H_1} e^{-i\beta_j H_2}) |\psi(0)\rangle, \quad (29)$$

where $\gamma_j = \Delta_t A(t_j)$ and $\beta_j = \Delta_t B(t_j)$.

Now, we address the original formulation of the QAOA, as shown in Algorithm 1 [9]. The Hamiltonian H_1 is usually set as $\sum_{i=1}^N |+\rangle$ for ease of preparation, and this is an unbiased initial state for optimization. The final Hamiltonian H_2 is our considered problem, mostly in the Ising formulation.

Algorithm 1 : Quantum Approximate Optimization Algorithm.

- 1: **Input:** The Hamiltonian states H_1 and H_2 , the number of layers p and an initial vector (γ, β) where $\gamma = (\gamma_1, \dots, \gamma_p)$ and $\beta = (\beta_1, \dots, \beta_p)$.
 - 2: Employ a classical optimizer for optimizing and updating parameters iteratively.
 - 3: **repeat**
 - 4: Set $|\psi(\gamma, \beta)\rangle = \prod_{j=p}^1 e^{-i\gamma_j H_1} e^{-i\beta_j H_2} |\psi(0)\rangle$ as the considered state.
 - 5: Estimate the expectation value of the Hamiltonian as $E(\psi(\gamma, \beta)) = \langle \psi(\gamma, \beta) | H_2 | \psi(\gamma, \beta) \rangle$.
 - 6: A classical optimization is used to adjust (γ, β) to minimize $E(\psi(\gamma, \beta))$.
 - 7: **until** convergence to the minimizer
 - 8: **Output:** The algorithm then returns the best solution for the Hamiltonian H_2 .
-

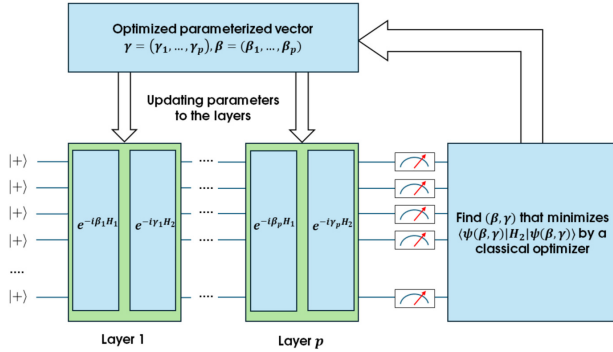


Fig. 4: An illustration of the quantum ansatz of QAOA with the initial state H_1 and the considered Hamiltonian H_2 .

To demonstrate the QAOA methodology in quantum circuit, let us assume that the Hamiltonian is in the Ising formulation (21). Regarding the j -th layer of the QAOA as $e^{-i\gamma_j H_1} e^{-i\beta_j H_2}$, the latter layer can be rewritten as

$$e^{-i\beta_j H_2} = \prod_{k,l} e^{i\beta_j J_{k,l} Z_k Z_l} \prod_k e^{i\beta_j J_k Z_k}. \quad (30)$$

The operations of $e^{i\beta_j J_k Z_k}$ can be implemented directly with the rotation gates $R_z(-2\beta_j J_k)$. We also notice that $e^{i\beta_j J_{k,l} Z_k Z_l} = e^{i\beta_j J_{k,l}}$ if the states of qubits k and l are

the same, and $e^{-i\beta_j J_{k,l}}$ otherwise. Hence, the circuit of this operation can be built in Figure 5. Afterwards, we can convert any Ising model to quantum circuits, and they are ready for the QAOA.

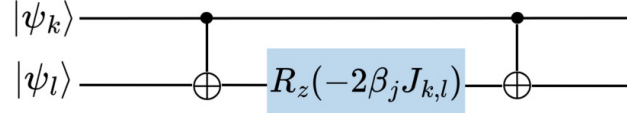


Fig. 5: The quantum gate implementation of $e^{i\beta_j J_{k,l} Z_k Z_l}$ held by a R_z gate and two CNOT gates.

From these developments, we have exploited the fundamentals of quantum computing, from definitions of qubits, quantum gates and circuits to the methodologies and concepts used in quantum optimization such as the QUBO and Simplex algorithm, coming with the Quantum Adiabatic Theorem and the original formulation of QAOA. We now have all the preliminaries ready for our proposed solution – the AO-based quantum-classical optimization coupled with the QAOA approach for the QUBO formation of the JSRP.

IV. PROPOSED AO-BASED HYBRID QUANTUM-CLASSICAL OPTIMIZATION SOLUTION

It is worth mentioning that the variables of \mathbf{x} are binary and the constraints (7b), (7e) are also binary constraints. Therefore, problem (7) is a mixed-integer problem (MIP) and also an NP-hard problem. In this section, we introduce an efficient AO approach for the proposed problem. Specifically, for fixed ship speed $\mathbf{v}^{(t)}$, the problem becomes a binary programming problem with vector of variables \mathbf{x} and can be converted to a QUBO formulation. Then, this expression will be transformed into a Hamiltonian representation for execution on a quantum processor. For the solved decision variables $\mathbf{x}^{(t+1)}$, the proposed problem becomes a convex optimization problem and can be solved directly.

A. Ship Routing Construction

For given feasible ships speed $\mathbf{v}^{(t)} \triangleq (v_k^{(t)}), k \in \mathcal{K}$, problem (7) can be expressed as follows:

$$\min_{\mathbf{x}} \sum_{k \in \mathcal{K}} C_k (\mathbf{x}_k, v_k^{(t)}), \quad (31a)$$

$$\text{s.t. } \sum_{i \in \mathcal{L}} x_{ki} = 1, \quad \forall k \in \mathcal{K}, \quad (31b)$$

$$\sum_{k \in \mathcal{K}} x_{ki} = 1, \quad \forall i \in \mathcal{L}, \quad (31c)$$

$$x_{ki} \in \{0, 1\}, \quad \forall k \in \mathcal{K}, i \in \mathcal{L}. \quad (31d)$$

This subproblem is a binary programming (BP) problem, and we would like to convert this problem into a QUBO form. To accomplish that, we employ the penalty parameters λ_1 and λ_2 to reformulate (31) as follows:

$$\begin{aligned}
\min_{\mathbf{x}} \quad & \sum_{k \in \mathcal{K}} C_k \left(\mathbf{x}_k, v_k^{(t)} \right) + \lambda_1 \sum_{k \in \mathcal{K}} \left(\sum_{i \in \mathcal{L}} x_{ki} - 1 \right)^2 \\
& + \lambda_2 \sum_{i \in \mathcal{L}} \left(\sum_{k \in \mathcal{K}} x_{ki} - 1 \right)^2, \\
\text{s.t.} \quad & x_{ki} \in \{0, 1\}, \forall k \in \mathcal{K}, i \in \mathcal{L}.
\end{aligned} \tag{32a}$$

$$\tag{32b}$$

In this QUBO expression, λ_1 and λ_2 are set at values that are sufficiently large to exaggerate the objective if the constraints are invalidated. As a result, the optimal solution to this problem should not be affected by penalties, ensuring that the constraints are satisfied.

To construct the \mathbf{Q} matrix, we first define the components of the \mathbf{Q} matrix due to the relationship between the variables. Let us denote $q_{(k_1, i_1), (k_2, i_2)}$ as the interaction between two binary variables $x_{k_1 i_1}$ and $x_{k_2 i_2}$. The linear terms at the diagonal entries of the matrix \mathbf{Q} can be calculated as

$$q_{(k, i), (k, i)} = S_{ki} \left(v_k^{(t)} \right) - \lambda_1 - \lambda_2, \tag{33}$$

where $-\lambda_1$ and $-\lambda_2$ are the penalties related to the shipment routing settings. Then, the quadratic terms that describe the interactions between the binary variables at the non-diagonal entries of matrix \mathbf{Q} are given as

$$q_{(k_1, i_1), (k_2, i_2)} = \begin{cases} 2\lambda_1, & k_1 = k_2 \wedge i_1 \neq i_2, \\ 2\lambda_2, & k_1 \neq k_2 \wedge i_1 = i_2, \\ 0, & k_1 \neq k_2 \wedge i_1 \neq i_2. \end{cases} \tag{34}$$

Herein, $2\lambda_1$ is the penalty applied when the ship k_1 transports two different cargoes i_1 and i_2 , ensuring that each ship transfers exactly one cargo at one pick-up node. The term $2\lambda_2$ enforces a penalty whenever a cargo is delivered by two different ships. Afterwards, other off-diagonal components are set as '0' because they have no interactions with each other.

The next step is to transform the QUBO problem into the Hamiltonian form. To address this, we introduce the spin variables $z_{ki} \in \{-1, 1\}$ and convert the binary decision variable x_{ki} to z_{ki} by the equation

$$x_{ki} = \frac{1 - z_{ki}}{2}. \tag{35}$$

By replacing (35) into the first component of (32a), we have

$$\begin{aligned}
& \sum_{k \in \mathcal{K}} C_k \left(\mathbf{x}_k, v_k^{(t)} \right) \\
& = \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{L}} x_{ki} S_{ki} \left(v_k^{(t)} \right) \\
& = \sum_{k \in \mathcal{K}, i \in \mathcal{L}} \frac{1 - z_{ki}}{2} S_{ki} \left(v_k^{(t)} \right) \\
& = \frac{1}{2} \sum_{\substack{k \in \mathcal{K} \\ i \in \mathcal{L}}} S_{ki} \left(v_k^{(t)} \right) - \frac{1}{2} \sum_{\substack{k \in \mathcal{K} \\ i \in \mathcal{L}}} z_{ki} S_{ki} \left(v_k^{(t)} \right).
\end{aligned} \tag{36}$$

Next, the second component of (32a) is reformulated as

$$\begin{aligned}
& \lambda_1 \sum_{k \in \mathcal{K}} \left(\sum_{i \in \mathcal{L}} x_{ki} - 1 \right)^2 = \lambda_1 \sum_{k \in \mathcal{K}} \left(\sum_{i \in \mathcal{L}} \frac{1 - z_{ki}}{2} - 1 \right)^2 \\
& = \frac{\lambda_1}{4} \sum_{k \in \mathcal{K}} \left(n - 2 - \sum_{i \in \mathcal{L}} z_{ki} \right)^2 \\
& = \frac{\lambda_1 n(n^2 - 3n + 4)}{4} - \frac{\lambda_1 (n - 2)}{2} \sum_{\substack{k \in \mathcal{K} \\ i \in \mathcal{L}}} z_{ki} \\
& \quad + \frac{\lambda_1}{2} \sum_{k \in \mathcal{K}} \sum_{\substack{i, j \in \mathcal{L} \\ i < j}} z_{ki} z_{kj}.
\end{aligned} \tag{37}$$

Finally, the third component of (32a) can be uniformly expressed as

$$\begin{aligned}
& \lambda_2 \sum_{i \in \mathcal{L}} \left(\sum_{k \in \mathcal{K}} x_{ki} - 1 \right)^2 = \frac{\lambda_2 n(n^2 - 3n + 4)}{4} \\
& \quad - \frac{\lambda_2 (n - 2)}{2} \sum_{\substack{k \in \mathcal{K} \\ i \in \mathcal{L}}} z_{ki} + \frac{\lambda_2}{2} \sum_{i \in \mathcal{L}} \sum_{\substack{k, j \in \mathcal{K} \\ k < j}} z_{ki} z_{ji}.
\end{aligned} \tag{38}$$

After transforming every component of the QUBO objective, we gather the Hamiltonian form of (32a) by adding (36), (37), (38) and taking out the constant terms. The final Hamiltonian form can be expressed as follows:

$$\begin{aligned}
H_{\text{final}} = & \sum_{\substack{k \in \mathcal{K} \\ i \in \mathcal{L}}} z_{ki} \left(\frac{-S_{ki} \left(v_k^{(t)} \right) - (\lambda_1 + \lambda_2) (n - 2)}{2} \right) \\
& + \frac{\lambda_1}{2} \sum_{k \in \mathcal{K}} \sum_{i, j \in \mathcal{L}, i < j} z_{ki} z_{kj} + \frac{\lambda_2}{2} \sum_{i \in \mathcal{L}} \sum_{k, j \in \mathcal{K}, k < j} z_{ki} z_{ji}.
\end{aligned} \tag{39}$$

Eventually, we design the QAOA to solve this subproblem directly, as shown in Algorithm 2. We start the process by entering some required inputs to the algorithm, that is, the known parameters in the mathematical model, penalty terms, feasible $\mathbf{v}^{(t)}$ for the system, and suitable quantum backend settings. Then, we design a parameterized quantum circuit (ansatz) with the known Hamiltonian objective and M layers for measurements. Associated with each layer m is an initial parameter vector (γ_m, δ_m) , and we set $(\gamma, \delta) \triangleq (\gamma_m, \delta_m)_{m=1, 2, \dots, M}$. A classical optimizer is also used to optimize the parameters (γ, δ) to minimize the objective value. The algorithm optimizes and updates the parameters until it attains convergence or reaches the maximum number of iterations.

B. Ship Speed Optimization

Since the optimal $\mathbf{x}^{(t+1)}$ has been obtained from Algorithm 2, we rewrite (7) as

$$\min_{\mathbf{v}} \sum_{k \in \mathcal{K}} C_k \left(\mathbf{x}_k^{(t+1)}, v_k \right), \tag{40a}$$

$$\text{s.t. } v_{\min} \leq v_k \leq v_{\max}, \quad \forall k \in \mathcal{K}, \tag{40b}$$

$$T_k(\mathbf{x}_k^{(t+1)}, v_k) \leq T_{\max}, \quad \forall k \in \mathcal{K}. \tag{40c}$$

Algorithm 2 : The designed QAOA for solving (31).

- 1: **Input:** Required parameters $n, v_{\min}, v_{\max}, o_i, o_{i+n}, K, p, q, w_k, m_k, P_{\text{fuel}}, \alpha, \beta, F, T_{\max}$, locations of nodes for problem (7); feasible $\mathbf{v}^{(t)}$, quantum backend settings; classical optimizer.
 - 2: Build a parameterized ansatz circuit with M layers and an initial vector. (γ, δ)
 - 3: Employ a classical optimizer for optimizing and updating parameters iteratively.
 - 4: Apply the parameterized vector (γ, δ) into the ansatz circuit.
 - 5: **repeat**
 - 6: Use the chosen classical optimizer to optimize and update (γ, δ)
 - 7: Measure the expectation value of the quantum state $\langle \gamma, \delta | H_{\text{final}} | \gamma, \delta \rangle$ and seek for convergence.
 - 8: **until** convergence
 - 9: **Output:** Optimized binary variables for route decision $\mathbf{x}^{(t+1)}$ and the optimal value of the objective.
-

From constraint (40c), we can find the lower bound of v_k as

$$v_k \geq \frac{\sum_{i \in \mathcal{L}} x_{ki}^{(t+1)} (d_{d(k),i} + d_{i,i_k+n} + d_{i+n,d(k)})}{T_{\max} - \sum_{i \in \mathcal{L}} x_{ki}^{(t+1)} (o_i + o_{i+n})} \triangleq v_k^{\text{low}}, \quad (41)$$

Therefore, constraints (40b) and (40c) can be combined as

$$\max \{v_{\min}, v_k^{\text{low}}\} \leq v_k \leq v_{\max}. \quad (42)$$

Considering the objective (40a) of this subproblem, we can see that this objective is a convex function. The objective can be rewritten as follows:

$$\sum_{k \in \mathcal{K}} C_k(\mathbf{x}_k^{(t+1)}, v_k) = \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{L}} \frac{x_{ki}^{(t+1)}}{v_k} (R_{ki} + Q_{ki} v_k^q), \quad (43)$$

where $R_{ki} \triangleq d_{d(k),i} \left(P_{\text{fuel}} K m_k^{\frac{2}{3}} p + \alpha w_i + F \right) + d_{i+n,d(k)} \times \left(P_{\text{fuel}} K m_k^{\frac{2}{3}} p + F \right) + \left(P_{\text{fuel}} K (w_i + m_k)^{\frac{2}{3}} p + \beta w_i + F \right) \times d_{i,i+n}$, and $Q_{ki} \triangleq \left(d_{d(k),i} m_k^{\frac{2}{3}} + d_{i,i+n} (w_k + m_k)^{\frac{2}{3}} + m_k^{\frac{2}{3}} \times d_{i+n,d(k)} \right) P_{\text{fuel}} K$. By solving the root of the first derivative of $C_k(\mathbf{x}_k^{(t+1)}, v_k)$, we have the desired value v_k^* for each ship k as

$$v_k^* = \left(\frac{\sum_{i \in \mathcal{L}} R_{ki} x_{ki}^{(t+1)}}{(q-1) \sum_{i \in \mathcal{L}} Q_{ki} x_{ki}^{(t+1)}} \right)^{\frac{1}{q}}. \quad (44)$$

Due to the convexity of $\sum_{k \in \mathcal{K}} C_k(\mathbf{x}_k^{(t+1)}, v_k)$ and the bounded constraint (42), the optimal speed $v^{(t+1)}$ is attained as

$$v_k^{(t+1)} = \begin{cases} \max \{v_{\min}, v_k^{\text{low}}\}, & v^* \leq \max \{v_{\min}, v_k^{\text{low}}\}, \\ v_{\max}, & v^* \geq v_{\max}, \\ v^*, & \max \{v_{\min}, v_k^{\text{low}}\} \leq v^* \leq v_{\max}. \end{cases} \quad (45)$$

C. Proposed Algorithm

Let $C_{\text{total}}^{(t)} \triangleq \sum_{k \in \mathcal{K}} C_k(\mathbf{x}_k^{(t)}, v_k^{(t)})$. From the Hamiltonian form of the problem and the optimal ships speed achieved in previous sections, the MIP problem (7) can be approximately solved using the proposed AO method shown in **Algorithm 3**.

Algorithm 3 : Proposed Algorithm for solving (7).

- 1: **Initialization:** Set $t = 0$, maximum number of iteration, N_{\max} ; generate the initial feasible points $\mathbf{x}^{(0)}, \mathbf{v}^{(0)}$, and choose the initial parameters for (7).
 - 2: **while** $(C_{\text{total}}^{(t)} = C_{\text{total}}^{(t-1)} \text{ or } t > N_{\max})$ **do**
 - 3: Execute Algorithm 2 to attain the optimal route scheduling variables $\mathbf{x}^{(t+1)}$;
 - 4: Follow (45) and use the given $\mathbf{x}^{(t+1)}$ to find the optimal ships speed $\mathbf{v}^{(t+1)}$;
 - 5: Calculate $C_{\text{total}}^{(t+1)}$ based on the optimal variables $\mathbf{x}^{(t+1)}$ and $\mathbf{v}^{(t+1)}$;
 - 6: Set $t = t + 1$.
 - 7: **end while**
 - 8: **Output:** optimal solutions $\{\mathbf{x}^*, \mathbf{v}^*\} = \{\mathbf{x}^{(t)}, \mathbf{v}^{(t)}\}$ and the minimized cost function.
-

V. PROPOSED QUBO OPTIMIZATION SOLUTION FOR JSRP

From the original problem (7) and the optimal speed determined at (44) and (45), the optimal value of $S_{ki}(v_k)$ can be properly calculated. Let us denote v_{ki}^* as the global minimum of $S_{ki}(v_k)$, which can be formulated as

$$v_{ki}^* = \left(\frac{R_{ki}}{(q-1)Q_{ki}} \right)^{\frac{1}{q}}, \quad (46)$$

and the minimum solution of $S_{ki}(v_k)$ under constrained is

$$v_{ki}^{\text{opt}} = \begin{cases} \max \{v_{\min}, v_k^{\text{low}}\}, & v_{ki}^* \leq \max \{v_{\min}, v_k^{\text{low}}\}, \\ v_{\max}, & v_{ki}^* \geq v_{\max}, \\ v_{ki}^*, & \max \{v_{\min}, v_k^{\text{low}}\} \leq v_{ki}^* \leq v_{\max}. \end{cases} \quad (47)$$

Next, we denote $S_{ki}^{\min} \triangleq S_{ki}(v_{ki}^{\text{opt}})$. The original problem (7) can now be rewritten as follows:

$$\min_{\mathbf{x}} \sum_{\substack{k \in \mathcal{K} \\ i \in \mathcal{L}}} x_{ki} S_{ki}^{\min}, \quad (48a)$$

$$\text{s.t. } \sum_{k \in \mathcal{K}} x_{ki} = 1, \quad \forall i \in \mathcal{L}, \quad (48b)$$

$$x_{ki} \in \{0, 1\}. \quad \forall k \in \mathcal{K}, i \in \mathcal{L}. \quad (48c)$$

This subproblem is a binary problem (BP), and again this problem should be converted to a QUBO problem (20). The penalties for this problem can be rewritten as follows:

$$\min_{\mathbf{x}} \sum_{\substack{k \in \mathcal{K} \\ i \in \mathcal{L}}} x_{ki} S_{ki}^{\min} + \lambda_3 \sum_{i \in \mathcal{L}} \left(\sum_{k \in \mathcal{K}} x_{ki} - 1 \right)^2, \quad (49a)$$

$$\text{s.t. } x_{ki} \in \{0, 1\}, \forall k \in \mathcal{K}, i \in \mathcal{L}. \quad (49b)$$

In this QUBO expression, λ_3 is set at sufficiently large values to exaggerate the objective if the constraints are invalidated. As a result, the optimal solution to this problem should not be affected by the penalties, ensuring that the constraints are satisfied. Likewise, the linear terms at the diagonal entries of the matrix \mathbf{Q} can be calculated as

$$q_{(k,i),(k,i)} = S_{ki}^{min} - \lambda_3, \quad (50)$$

where $-\lambda_3$ is the penalty related to the shipment routing settings. Then, the quadratic terms that describe the interactions between the binary variables at the non-diagonal entries of matrix \mathbf{Q} are given as

$$q_{(k_1,i_1),(k_2,i_2)} = \begin{cases} 2\lambda_3, & k_1 \neq k_2 \wedge i_1 = i_2, \\ 0, & i_1 \neq i_2. \end{cases} \quad (51)$$

Herein, the term $2\lambda_3$ enforces a penalty whenever a cargo is delivered by two different ships. Afterwards, other off-diagonal components are set as '0' because they have no interactions with each other.

The next step is to transform the QUBO problem into the Hamiltonian form. The spin variables from (35) are used again to convert our QUBO problem into a new Hamiltonian form. By replacing (35) into the first component of (49), we have

$$\sum_{\substack{k \in \mathcal{K} \\ i \in \mathcal{L}}} x_{ki} S_{ki}^{min} = \frac{1}{2} \sum_{\substack{k \in \mathcal{K} \\ i \in \mathcal{L}}} S_{ki}^{min} - \frac{1}{2} \sum_{\substack{k \in \mathcal{K} \\ i \in \mathcal{L}}} z_{ki} S_{ki}^{min}. \quad (52)$$

Then, the other component of (32a) can be uniformly expressed as

$$\begin{aligned} \lambda_3 \sum_{i \in \mathcal{L}} \left(\sum_{k \in \mathcal{K}} x_{ki} - 1 \right)^2 &= \frac{\lambda_3 n(n^2 - 3n + 4)}{4} \\ &- \frac{\lambda_3(n-2)}{2} \sum_{\substack{k \in \mathcal{K} \\ i \in \mathcal{L}}} z_{ki} + \frac{\lambda_3}{2} \sum_{i \in \mathcal{L}} \sum_{\substack{k, j \in \mathcal{K} \\ k < j}} z_{ki} z_{ji}. \end{aligned} \quad (53)$$

After converting every component of the QUBO objective, we gather the Hamiltonian form of (32a) by adding (52), (53) and taking out the constant term. The final Hamiltonian form can be sufficiently expressed as follows:

$$\begin{aligned} H_{\text{final}} &= \sum_{\substack{k \in \mathcal{K} \\ i \in \mathcal{L}}} z_{ki} \left(\frac{-S_{ki}^{min} - \lambda_3(n-2)}{2} \right) \\ &+ \frac{\lambda_3}{2} \sum_{i \in \mathcal{L}} \sum_{\substack{k, j \in \mathcal{K} \\ k < j}} z_{ki} z_{ji}. \end{aligned} \quad (54)$$

Generally, we can search the minimal solution of this Hamiltonian formulation with the same process of Algorithm 2. The main difference is the considered Hamiltonian, and we only need to run the algorithm once. The process only stops until we get the convergence or the maximum number of iterations are exceeded.

VI. SIMULATION RESULTS AND DISCUSSIONS

A. Parameter Settings

The parameters of this paper are referred to in [29], [19]. For simulations, we examine the cases of $n = 3$ and $n = 4$. The distances between the nodes (in nautical miles) are presented in Table III and Table II. The maximum operation time is a week, equal to $T_{\max} = 168$ hours. The speed of the ship ranges from $v_{\min} = 6$ knots to $v_{\max} = 20$ knots. The fuel consumption constants are modeled as $K = 1.8 \times 10^{-5}$, $p = 0$, and $q = 3$. The fuel price is assumed to be $P_{\text{fuel}} = 300$ USD/ton. The unit cargo port inventory cost is set to $\alpha = 2$ USD/ton/day, and the unit cargo in-transit inventory cost is $\beta = 1.5$ USD/ton/day. The freight rate can be assumed as $F = 10,000$ USD/day. Each cargo is assumed to have the same weight as $w_i = 9,000$ tons, $\forall i \in \mathcal{L}$, and the weights of the ships without any cargoes are arranged as $\mathbf{m} \triangleq \{m_k\}_{k=1,\dots,n} = [5000, 4500, 4750, 4500]$ tons. The execution times in the loading ports are modeled as $o_i = 15$ hours and $o_{i+n} = 17$ hours, $\forall i \in \mathcal{L}$.

Since our problem is to deal with binary variables x_{ki} , the number of qubits we have to encounter is n^2 . The simulations are constructed and implemented in a Python environment with packages such as `qiskit`, `qiskit_algorithms`. We also use `matplotlib` to visualize the numerical results obtained by Aer Simulator, IBM computer or Brute-Force further to examine the validity and robustness of these methodologies. For quantum hardware settings, the code is run via `ibm_quebec` backend with 127 qubits maximum per iteration.

TABLE II: Distance Matrix with $n = 3$ (in nautical miles).

From - To	Distance (nautical miles)
From Depots to Loading Ports:	
$d_{(1,1)}, d_{(1,2)}, d_{(1,3)}$	720.0, 550.0, 610.0
$d_{(2,1)}, d_{(2,2)}, d_{(2,3)}$	470.0, 520.0, 580.0
$d_{(3,1)}, d_{(3,2)}, d_{(3,3)}$	650.0, 550.0, 480.0
From Loading Ports to Discharging Ports:	
$d_{(1,4)}, d_{(2,5)}, d_{(3,6)}$	600.0, 700.0, 800.0
From Discharging Ports to Depots:	
$d_{(4,1)}, d_{(4,2)}, d_{(4,3)}$	600.0, 650.0, 710.0
$d_{(5,1)}, d_{(5,2)}, d_{(5,3)}$	720.0, 730.0, 690.0
$d_{(6,1)}, d_{(6,2)}, d_{(6,3)}$	580.0, 740.0, 715.0

TABLE III: Distance Matrix with $n = 4$ (in nautical miles).

From - To	Distance (nautical miles)
From Depots to Loading Ports:	
$d_{(1,1)}, d_{(1,2)}, d_{(1,3)}, d_{(1,4)}$	720.0, 550.0, 610.0, 600.0
$d_{(2,1)}, d_{(2,2)}, d_{(2,3)}, d_{(2,4)}$	470.0, 520.0, 580.0, 650.0
$d_{(3,1)}, d_{(3,2)}, d_{(3,3)}, d_{(3,4)}$	650.0, 550.0, 480.0, 590.0
$d_{(4,1)}, d_{(4,2)}, d_{(4,3)}, d_{(4,4)}$	610.0, 450.0, 520.0, 580.0
From Loading Ports to Discharging Ports:	
$d_{(1,5)}, d_{(2,6)}, d_{(3,7)}, d_{(4,8)}$	600.0, 700.0, 800.0, 900.0
From Discharging Ports to Depots:	
$d_{(5,1)}, d_{(5,2)}, d_{(5,3)}, d_{(5,4)}$	600.0, 650.0, 710.0, 740.0
$d_{(6,1)}, d_{(6,2)}, d_{(6,3)}, d_{(6,4)}$	720.0, 730.0, 690.0, 620.0
$d_{(7,1)}, d_{(7,2)}, d_{(7,3)}, d_{(7,4)}$	580.0, 740.0, 715.0, 820.0
$d_{(8,1)}, d_{(8,2)}, d_{(8,3)}, d_{(8,4)}$	800.0, 745.0, 675.0, 620.0

To strengthen the constraints in the QUBO formulation,

the penalty parameters λ_1 , λ_2 and λ_3 should be larger than the maximum value of the objective (7). To ensure this, the ideal penalties of our QUBO formulation can be set as $\lambda_1 = \lambda_2 = \lambda_3 = \sum_{k \in \mathcal{K}} \max_{i \in \mathcal{L}} S_{ki}(v_{\max})$. This setting ensures that the maximum value of (7) cannot exceed λ_1 , λ_2 and λ_3 , and the penalties themselves are not too overwhelming with the objectives (7a) and (49a).

B. Numerical Results and Discussions

1) A Comparative Analysis of the Convergence Behavior of the Algorithms

Fig. 6a and 6b represent the convergence dynamics in the case of a 9-variable model using the Qiskit Aer simulator with the COBYLA optimizer and the quantum hardware provided by IBM. Initially, in the Proposed AO, the IBM quantum hardware displays oscillations in around 40 iterations as the exploratory phase. The Aer Simulator, however, spends roughly 60 iterations in the searching phase, but both platforms take around 10 iterations more to seek for convergence. The algorithms show a high degree of fluctuation, which may be caused by updates to the QAOA parameters (γ, δ) , or the optimizer is struggling to find an optimal direction. In the last iterations, the simulator is more stable and begins to converge gradually, indicating that the algorithm likely reaches near-optimal solutions to find the low-energy state. The similar pattern exists in the JSRP Algorithm, where the IBM Computer and Aer Simulator spend 40 and 70 iterations for the searching phase, respectively. It should be noted that the minimal value derived by the quantum hardware is slightly greater than that of the Aer Simulator. This difference is mainly caused by the noise in real IBM hardware, including physical limitations such as gate errors, measurement errors and decoherence, thereby generating errors in the computational part. Despite this small lack of accuracy, with fewer iterations, the IBM computer emerges in reaching a close solution to the Aer Simulator and successfully validating the effectiveness of the Proposed AO and JSRP algorithms.

Fig. 7a and 7b illustrate the convergence behavior in the scenario of 16 qubits achieved between Aer Simulator and IBM Computer. Regarding the Proposed AO Algorithm, Aer Simulator describes the QUBO objective function (32) after roughly 70 iterations exploring the optimal solution and gradually converges to the optimal point in around 10 iterations. In contrast, the results executed by IBM Computer indicate that the searching process takes only 40 iterations and around 15 iterations more to reach the optimal value, which is likely the same as the results in the 9-qubit case. For the JSRP Algorithm, the convergence behavior is equivalent to Fig. 6b, where the simulator takes 70 iterations approximately to find the optimal point and 20 last iterations to reach convergence. Meanwhile, the quantum hardware set up by IBM spends around 40 iterations to the searching process and nearly 20 iterations to attain the optimal point at convergence. It can be noticed that the gap between the hardware results and the simulator results is larger compared to the 9-qubit case, which probably can be caused by the increase in quantum noises when handling with more qubits.

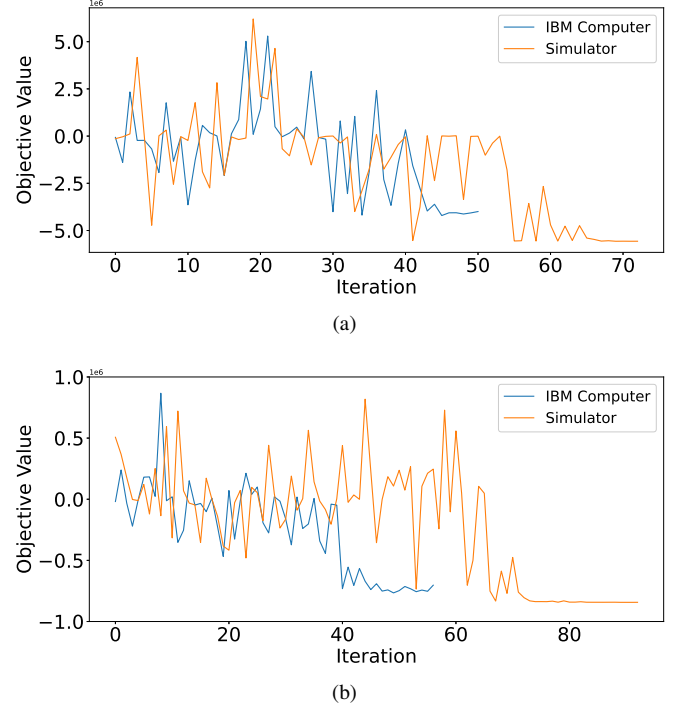


Fig. 6: Convergence behavior of the QAOA-based Algorithm in the scenario of 9 qubits. (a) First iteration on Proposed AO Algorithm, (b) JSRP Algorithm.

2) A comparison using numerical results of various methods to minimize the total cost:

Fig. 8 illustrates the comparative results of the total cost from different methodologies. The Fixed Speed algorithm is optimized with fixed $v_k = \frac{v_{\min} + v_{\max}}{2}, \forall k \in \mathcal{K}$, whereas Fixed Routing is executed with binary decision variables set as $x_{14} = x_{21} = x_{32} = x_{43} = 1$, and '0' otherwise. The simulation shows that the Fixed Speed method has the highest total cost of over \$600,000, while the Fixed Routing has the second one with \$596,785.60. Nevertheless, it cannot be concluded that the Fixed Speed algorithm always performs better than the Fixed Routing due to varying settings. It is also recognized that the Proposed AO, the QUBO optimization for JSRP and Brute-Force methods yield the same total cost as \$576,512.91, proving the effectiveness and robustness of the proposed algorithm. Since Fixed Speed and Fixed Routing approaches merely consider one variable for total cost minimization, resulting in underperformance compared to the Proposed AO, QUBO JSRP and the Brute-Force algorithms.

3) Routing model solution in the case of $n = 3$

Given in Fig. ?? is the graph illustrating the shipment routing solution with three blue depots, denoted as (D_1, D_2, D_3) , three green loading ports (1, 2, 3) and three red delivery nodes (4, 5, 6). This visualization helps demonstrate the routing structure and the travel distances between the nodes in the system model. Initially, the routing decision of the AO method is to gather $(D_i \rightarrow i \rightarrow i + n)$ for each $i = 1, 2, \dots, n$. The result shows that to derive the optimized solution, ship-1 and ship-2 have to change their carriers to each other and assign the associated optimal speed in this case. This problem only

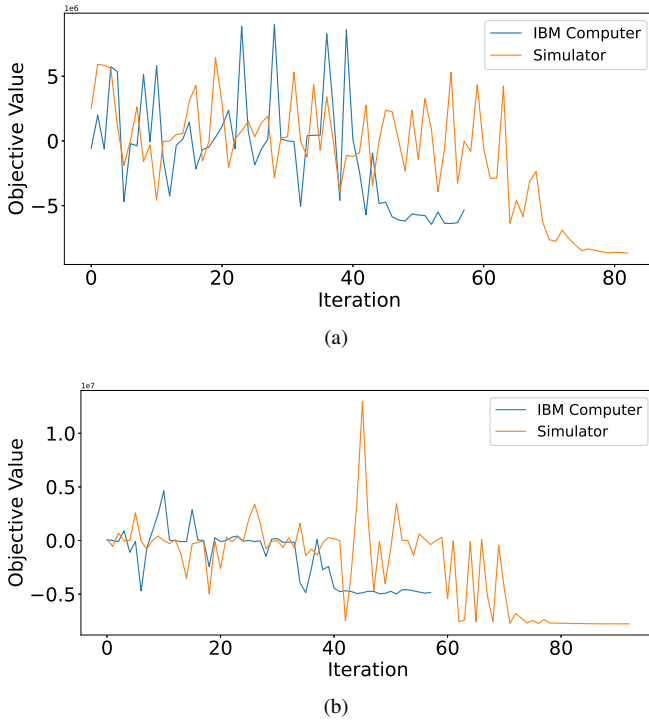


Fig. 7: Convergence behavior of the QAOA-based Algorithm in the scenario of 16 qubits. (a) First iteration on Proposed AO Algorithm, (b) JSRP Algorithm.

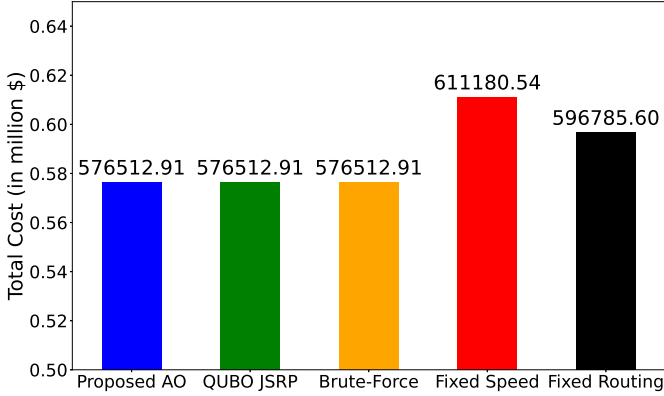


Fig. 8: Simulations of different methods compared with the proposed AO algorithm to minimize the total cost

costs one iteration for the proposed AO approach to achieve and one more iteration to check the convergence, declaring that the multidimensional optimization problems can be potentially solved by the AO algorithm.

4) Running time comparison between the JSRP solved by QUBO and Brute-Force solution

Finally, Fig. 10 demonstrates the exponential growth of the running time ratio between the JSRP solved by QAOA and the Brute-Force algorithm considering the same increase of variables from 4 to 16. The figure shows that the Brute-Force approach has an immense increase in running time when the number of variables inclines, with 952.44 times higher for the rise of 16 variables compared to 8.77 times with the 4-to-9-variable case, respectively. Meanwhile, considering the

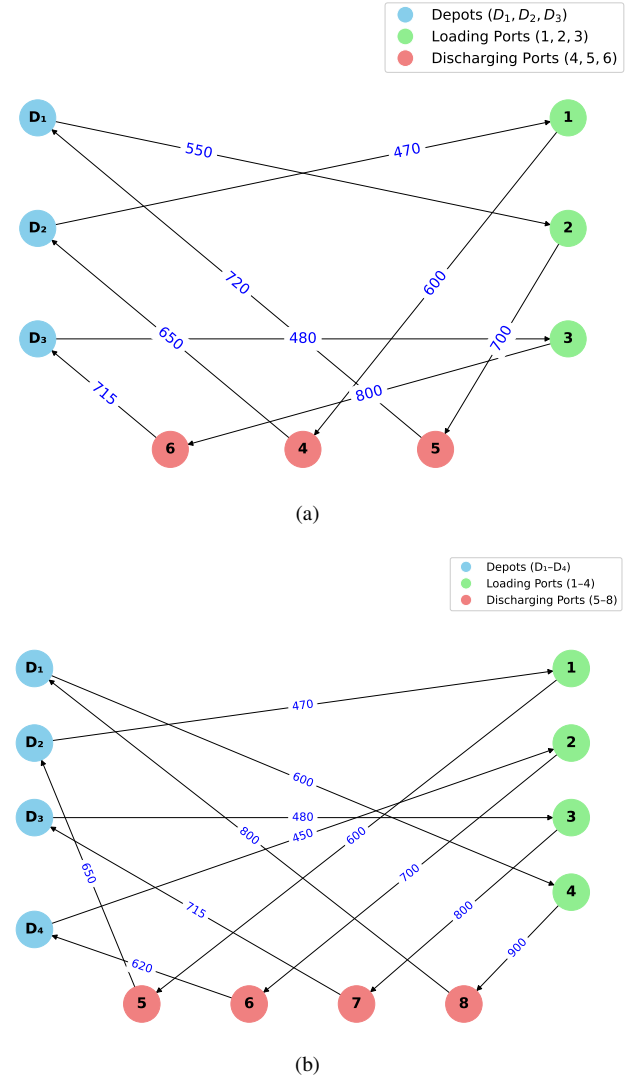


Fig. 9: A simulation on the shipment routing problem in the case of $n = [3, 4]$.

QAOA method for JSRP in the range of 4 to 16 variables, 182.60 times higher compared to 3.82 times of 9 variables is a quite moderate number. Particularly, these results reveal that the quantum-centric approach can derive near-optimal performance for sophisticated problems which seem classically intractable.

VII. CONCLUSION

In this study, we have investigated the integrated challenge of minimizing the total operational cost of maritime logistics, an NP-hard and computationally complex problem, through joint optimization of ship routing and speed. We presented comprehensive preliminaries of quantum computing and quantum optimization to facilitate a clear understanding of our proposed solutions. Two distinct optimization frameworks have been proposed: an AO-based solution and a JSRP QUBO-based transformation optimization solution. The AO-based solution separates the original problem into a routing subproblem, solved using the QAOA-based algorithm, and a speed

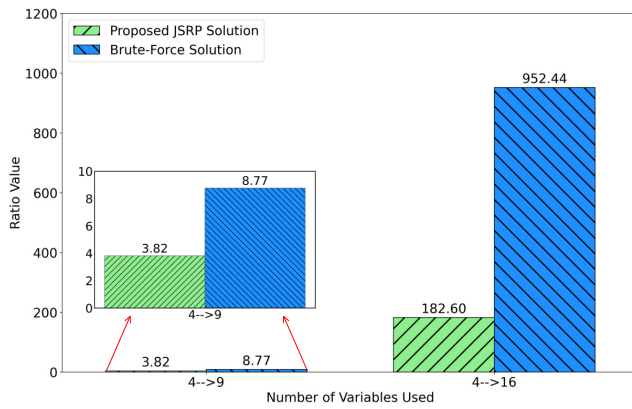


Fig. 10: The exponential growth of the running time ratio between the Pure QAOA method on JSRP compared with the classical Brute-Force method.

subproblem addressed by a closed-form solution. In contrast, the JSRP QUBO-based approach concurrently optimizes speed and routing decisions within a unified quantum optimization framework. Numerical simulations have demonstrated the superior performance and effectiveness of both proposed methods compared to classical benchmark approaches, achieving results equivalent to those obtained by the brute-force algorithm. These results highlight the considerable potential of hybrid quantum-classical techniques for solving complex logistic challenges in maritime transportation. Future research could explore advanced quantum-classical hybrid strategies, such as adaptive variational algorithms or error-mitigation techniques, to further improve the scalability and robustness of quantum optimization methods for large-scale maritime logistics.

REFERENCES

- [1] United Nations Conference on Trade and Development, *Review of Maritime Transport 2023: Towards a Green and Just Transition*. Geneva: United Nations, 2023. [Online]. Available: <https://unctad.org/publication/review-maritime-transport-2023>
- [2] J. Ksciuk, S. Kuhlemann, K. Tierney, and A. Koberstein, "Uncertainty in maritime ship routing and scheduling: A literature review," *European Journal of Operational Research*, vol. 308, no. 2, pp. 499–524, 2023.
- [3] L. Huang, C. Wan, Y. Wen, R. Song, and P. van Gelder, "Generation and application of maritime route networks: Overview and future research directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 1, pp. 620–637, 2025.
- [4] L. P. Perera, B. Mo, L. A. Kristjansson, P. C. Jonvik, and J. O. Svardal, "Evaluations on ship performance under varying operational conditions," *International Conference on Offshore Mechanics and Arctic Engineering*, vol. Volume 7: Ocean Engineering, p. V007T06A060, 05 2015.
- [5] Y. Han, W. Ma, and D. Ma, "Green maritime: An improved quantum genetic algorithm-based ship speed optimization method considering various emission reduction regulations and strategies," *Journal of Cleaner Production*, vol. 385, p. 135814, 2023.
- [6] W. Du, Y. Li, G. Zhang, C. Wang, B. Zhu, and J. Qiao, "Ship weather routing optimization based on improved fractional order particle swarm optimization," *Ocean Eng.*, vol. 248, p. 110680, 2022.
- [7] R. Mandelbaum, A. D. Córcoles, and J. Gambetta, "IBM's big bet on the quantum-centric supercomputer: Recent advances point the way to useful classical-quantum hybrids," *IEEE Spectrum*, vol. 61, no. 9, pp. 24–33, Sep. 2024.
- [8] R. Shaydulin *et al.*, "Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem," *Sci. Adv.*, vol. 10, no. 22, pp. 1–10, May 2024.
- [9] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.
- [10] D. J. Egger, J. Mareček, and S. Woerner, "Warm-starting quantum optimization," *Quantum*, vol. 5, p. 479, Jun. 2021.
- [11] C. H. V. Cooper, "Exploring potential applications of quantum computing in transportation modelling," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14712–14720, Sep. 2022.
- [12] S. J. Weinberg, F. Sanches, T. Ide, K. Kamiya, and R. Correll, "Supply chain logistics with quantum and classical annealing algorithms," *Sci. Reports*, vol. 13, no. 4770, pp. 24–33, Mar. 2023.
- [13] N. Mohanty, B. K. Behera, and C. Ferrie, "Analysis of the vehicle routing problem solved via hybrid quantum algorithms in the presence of noisy channels," *IEEE Trans. on Quantum Eng.*, vol. 4, pp. 1–14, Sep. 2023.
- [14] S. Harwood, C. Gambella, D. Trenev, A. Simonetto, D. Bernal, and D. Greenberg, "Formulating and solving routing problems on quantum computers," *IEEE Trans. on Quantum Eng.*, vol. 2, pp. 1–17, Jan. 2021.
- [15] H. Guo, J. Wang, J. Sun, and X. Mao, "Multi-objective green vehicle scheduling problem considering time window and emission factors in ship block transportation," *Scientific Reports*, vol. 14, 05 2024.
- [16] Z. Zhang, H. Qin, and Y. Li, "Multi-objective optimization for the vehicle routing problem with outsourcing and profit balancing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1987–2001, 2020.
- [17] Z. Xia, Z. Guo, W. Wang, and Y. Jiang, "Joint optimization of ship scheduling and speed reduction: A new strategy considering high transport efficiency and low carbon of ships in port," *Ocean Engineering*, vol. 233, p. 109224, 2021.
- [18] W. Ma, D. Ma, Y. Ma, J. Zhang, and D. Wang, "Green maritime: a routing and speed multi-objective optimization strategy," *Journal of Cleaner Production*, vol. 305, p. 127179, 2021.
- [19] M. Wen, D. Pacino, C. Kontovas, and H. Psaraftis, "A multiple ship routing and speed optimization problem under time, cost and environmental objectives," *Transp. Res. D Trans. Environ.*, vol. 52, pp. 302–321, May 2017.
- [20] A. De, J. Wang, and M. K. Tiwari, "Fuel bunker management strategies within sustainable container shipping operation considering disruption and recovery policies," *IEEE Transactions on Engineering Management*, vol. 68, no. 4, pp. 1089–1111, 2021.
- [21] C. A. Kontovas, "The green ship routing and scheduling problem (gsrsp): A conceptual approach," *Transportation Research Part D: Transport and Environment*, vol. 31, pp. 61–69, 2014.
- [22] Q. Li, Z. Huang, W. Jiang, Z. Tang, and M. Song, "Quantum algorithms using infeasible solution constraints for collision-avoidance route planning," *IEEE Transactions on Consumer Electronics*, 2024.
- [23] D. Alanis, P. Botsinis, Z. Babar, H. V. Nguyen, D. Chandra, S. X. Ng, and L. Hanzo, "Quantum-aided multi-objective routing optimization using back-tracing-aided dynamic programming," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7856–7860, 2018.
- [24] E. Osaba, E. Villar-Rodríguez, and A. Asla, "Solving a real-world package delivery routing problem using quantum annealers," *Scientific Reports*, vol. 14, 10 2024.
- [25] F. Li, A. R. Mazumder, and M. Zhao, "Quantum annealing approaches to solving the shipment rerouting problems," 2025. [Online]. Available: <https://arxiv.org/abs/2501.05624>
- [26] T. Yang, Z. Cui, C. Peng, J. Wu, F. Liu, and Y. Yang, "Integrated communication and computing maritime networks design for green metaverse," *IEEE Wireless Communications*, vol. 30, no. 5, pp. 120–126, 2023.
- [27] W. Zhao, H. Wang, J. Geng, W. Hu, Z. Zhang, and G. Zhang, "Multi-objective weather routing algorithm for ships based on hybrid particle swarm optimization," *Journal of Ocean University of China*, vol. 21, pp. 28–38, 02 2022.
- [28] D. Liu, J. Zhang, J. Cui, S.-X. Ng, R. G. Maunder, and L. Hanzo, "Deep learning aided routing for space-air-ground integrated networks relying on real satellite, flight, and shipping data," *Wireless Communications*, vol. 29, no. 2, p. 177–184, Apr. 2022.
- [29] H. N. Psaraftis and C. A. Kontovas, "Ship speed optimization: Concepts, models and combined speed-routing scenarios," *Transp. Res. Part C Emerg. Technol.*, vol. 44, pp. 52–69, Jul. 2014.
- [30] R. S. Sutor, *Dancing with Qubits: How quantum computing works and how it can change the world*, 1st ed. Packt, 2019.
- [31] A. Jacquier and O. Kondratyev, *Quantum Machine Learning and Optimisation in Finance*, 1st ed. Packt, 2022.
- [32] E. F. Combarro and S. González-Castillo, *A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms*, 1st ed. Packt, 2023.

- [33] A. Ambainis and O. Regev, “An elementary proof of the quantum adiabatic theorem,” *arXiv:quant-ph arXiv:0411152*, 2006.
- [34] H. F. Trotter, “On the product of semi-groups of operators,” *Proceedings of the American Mathematical Society*, vol. 10, no. 4, pp. 545–551, Aug. 1959.