Journal of Industrial and Systems Engineering
Vol. 17, Special issue: 20th Iranian International Industrial Engineering Conference
Autumn (October) 2025; pp. 48-57

Quantum Modeling of the Dynamic Ride-sharing Problem

Erfan Amani Bani[1], Kourosh Eshghi[1*]

[1] *Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran*

Abstract
Mathematical modeling and the subsequent development of optimization algorithms for problems have been the core focus of operations research scientists. However, the challenges of solving complex models within an acceptable time frame have consistently fostered creativity in this field. Quantum computing has been proposed as an alternative to binary computing for several decades. In recent years, scientists and researchers in operations research have paid significant attention to utilizing and integrating this logic with optimization. Specifically, quantum variants of many optimization algorithms have been developed; however, more focus needs to be placed on modeling optimization problems using quantum variables. In this paper, a practical problem, the dynamic ride-sharing problem, is redefined and subsequently modeled using quantum variables. The resulting model, defined based on quantum variables, is fully compatible with quantum algorithms.

Keywords: Dynamic ride-sharing problem; quantum computing; quantum optimization

1. Introduction and literature review

The expansion of online taxi services across the world over the past two decades is undoubtedly due to the Fourth Industrial Revolution and the spread of the Internet. These services currently act as an effective intermediary in connecting the supply side—drivers—with the demand side—passengers. Through fair pricing, they can satisfy both parties involved in this exchange. However, these services still account for only a small fraction of urban trips. The reason behind this is that the continuous use of such services, particularly for individuals traveling alone to a destination, is costly. This high cost limits the ability of these service providers to attract a larger market share.

One key solution to this issue is for passengers who may not necessarily know each other but are traveling from nearby locations to a common destination in another area—again, close to each other—to share a ride with a single driver. However, what exactly does this mean?

Ride-sharing is a transportation service in which individuals share a vehicle for a trip, either through pre-arranged carpooling or on-demand services facilitated by applications like Uber or Snapp. This service allows passengers heading in the same direction to split costs and reduce the number of vehicles on the road, contributing to greater convenience, environmental sustainability, and cost savings. Ride-sharing platforms use algorithms to match drivers with passengers based on location, optimizing routes and minimizing wait times. This service differs from traditional taxi services as it provides a more flexible and often cheaper alternative for both drivers and passengers [2].

This form of service, alongside existing taxi services, could significantly help these companies attract more of the market. The question, however, is why this service has received so little attention so far. The answer is that while online taxi service providers are interested in offering such a service, the challenges involved make its development financially unfeasible. One of the primary challenges is the optimal allocation of passengers to vehicles and then computing the optimal sequence of destinations that drivers should follow. Solving this challenge can accelerate drivers' acceptance of ride requests and reduce passenger wait times. Addressing this challenge using algorithms based on classical computational logic—binary computation—seems highly unlikely, as exact algorithms are time-consuming, and heuristic algorithms, as is well known, fail to solve the problem at real-world scales.

---

* Corresponding Author ISSN: 1735-8272,

The concept of ride-sharing has been discussed in transportation optimization literature for several decades. Over time, it has evolved into various modeling approaches that can be categorized into four main types: one-to-one (transporting a single passenger to a single destination), one-to-many (transporting a single passenger to multiple destinations), many-to-one (transporting multiple passengers to a single destination), and many-to-many (transporting multiple passengers to multiple destinations). Research in this field started with fundamental problem definitions and has progressively moved toward practical implementations and the development of innovative methods.

The concept of ride-sharing has been discussed in transportation optimization literature for several decades. Over time, it has evolved into various modeling approaches that can be categorized into four main types: one-to-one (transporting a single passenger to a single destination), one-to-many (transporting a single passenger to multiple destinations), many-to-one (transporting multiple passengers to a single destination), and many-to-many (transporting multiple passengers to multiple destinations). Research in this field started with fundamental problem definitions and has progressively moved toward practical implementations and the development of innovative methods.

1.1 Early Formulations of Ride-Sharing Problems

The foundation of ride-sharing was laid by [3]. They modeled the dial-a-ride problem with a single vehicle, where real-time many-to-many ride requests were considered, in two different ways: first as a static model and then as a dynamic model. In the static case, real-time requests that emerged during the trip were not considered. Their objective was to minimize the total service time for all customers and reduce dissatisfaction caused by delays. For this type of problem, they developed a dynamic programming approach that accounted for additional constraints, including vehicle capacity. In the dynamic case, [3] allowed real-time requests to be considered while enroute. In this scenario, a service sequence for customers was determined, but this sequence was not fixed in length and could be updated in real time.

Later, [4] extended the dynamic assignment problem—a fundamental topic in operations research—by incorporating stochastic elements. One of the key applications of stochastic dynamic assignment problems is in ride-sharing planning, as demand in such problems can be modeled as randomly arriving over time, requiring the allocation of vehicles for service. Subsequently, [5] introduced a more general version of the dynamic assignment problem and proposed a solution approach that could be applied to more complex ride-sharing scenarios, including many-to-many cases. They developed an algorithm based on a non-myopic policy to solve this dynamic programming problem.

1.2 Dynamic Problems and Computational Efficiency

As ride-sharing research advanced, its overlap with dynamic routing problems became evident. [6] modeled the dynamic pickup and delivery problem and developed a heuristic algorithm to solve it. They then extended this problem to vehicle routing under similar assumptions. [7] revisited the dynamic pickup and delivery problem but instead developed various versions of variable neighborhood search metaheuristics for its solution. Meanwhile, [8] introduced a graph-based partitioning approach in which a many-to-many ride-sharing system was decomposed into separate one-to-one systems, significantly reducing the computational cost of solving the problem.

1.3 Advancements in Solution Methods

To achieve more efficient solutions, researchers have introduced innovative methods:

- Heuristic and Metaheuristic Methods: Continuing along this path, [9] developed a decision-support system, where one of its main tasks was optimizing the assignment and sequencing of customer drop-offs. Their system employed a three-phase optimization approach: the first phase used a heuristic algorithm to make low-cost assignment decisions, the second phase filtered the best solutions from the first phase, and the third phase employed a simulation-based approach to evaluate each solution under different scenarios and determine the best decision. [10] introduced a novel problem definition by incorporating a new objective function that considered the geographical characteristics of each trip. They solved the problem using clustering-based heuristics, providing a fresh perspective on the problem.
- Exact and Decomposition-Based Methods: [11] initially developed a mathematical model to represent the one-to-one ride-sharing assignment problem with multiple hubs. Their solution approach involved two steps: first, reducing the problem size and then solving it using a custom-developed decomposition-based algorithm. [12] extended the work of [11] by addressing the challenge of participant engagement in the system. To tackle this, they proposed a trip-swapping mechanism that allowed a passenger to purchase additional seats in a vehicle, effectively making it a private ride. Their goal was to maximize user surplus, and their numerical results demonstrated that this mechanism effectively addressed the challenge. [13] further built on [11] by proposing an online optimization algorithm for solving the problem. A similar perspective to [10] was taken by [14], who redefined the problem by incorporating multiple simultaneous objectives in the objective function, such as

travel time, delays, and vehicle usage costs. They then solved the problem for large-scale instances using a branch-and-bound method.

1.4 Scientific Advancements, Recent Developments, and Future Outlook

Flexibility in system parameters has had a significant impact on performance. [15] approached the problem from a different angle. It is natural that in a ride-sharing system, some customers may not have their requests fulfilled. They introduced the assumption that a separate group of dedicated drivers was available to serve these unmet requests. They incorporated this assumption into their modeling and solved the problem using heuristic construction and improvement algorithms. Building on this, [16] demonstrated that in a simple ride-sharing system, increasing flexibility in aspects such as driver departure times could lead to higher matching rates, ultimately improving system performance.

In recent years, machine learning and reinforcement learning have transformed problem-solving approaches. For example, [17] explored optimization methods based on learning and demonstrated the potential of data-driven decision-making in dynamic ride-sharing. Specifically, they modeled a classic dynamic ride-sharing problem, but instead of using traditional solution methods, they developed a deep reinforcement learning architecture. By integrating this algorithm with integer-linear programming techniques, they achieved optimal solutions.

Researchers continue to seek solution methods that can solve these systems within a reasonable time frame for practical, real-world applications. For instance, [18] developed an adaptive large neighborhood search method to solve the problem efficiently. Compared to classical methods developed so far, their approach showed significant advantages, but it did not necessarily guarantee globally optimal solutions for real-world problem sizes. There is still no exact solution for this problem or other NP-hard optimization problems that can be solved in polynomial time. This very challenge forms the core idea of this dissertation.

Over the past two decades, alongside the evolution of dynamic ride-sharing literature, a transformation has been occurring in computational methods. Classical computing, based on bits, has been gradually giving way to quantum computing. In the following sections, we will review the literature on advancements in quantum computing and then explore the intersection of optimization and quantum computing.

1.5 Quantum modeling and solution methodologies

Quantum optimization algorithms have been extensively studied in recent years, particularly in the context of combinatorial optimization problems such as the Vehicle Routing Problem (VRP). These algorithms can be categorized into two main types: metaheuristic quantum algorithms and deterministic quantum algorithms.

Metaheuristic quantum algorithms aim to improve traditional heuristic methods like the Genetic Algorithm and Particle Swarm Optimization by integrating quantum computing principles. One of the earliest studies in this domain was conducted by [19], who introduced a quantum-inspired version of the Genetic Algorithm by leveraging the concept of parallel evolution across multiple solution populations. This approach was later refined by [20], who incorporated qubits instead of classical binary representation, leading to improved solution quality and faster convergence. Further advancements were made by [21] and [22], who utilized the chaotic properties of qubits to enhance the mutation operator and optimize the search space more effectively. Similarly, quantum principles were applied to Particle Swarm Optimization, where [23] introduced a method to update particle velocities using qubit substitution rules, while [24] used quantum chaos theory to enhance local and global search capabilities.

On the other hand, deterministic quantum algorithms focus on adapting classical exact optimization methods to quantum computing frameworks. One such example is the Quantum Simplex Algorithm proposed by [25], which modified key simplex steps—including optimality checks, unboundedness detection, and pivot element selection—using quantum operations to achieve significant reductions in computational complexity. Similarly, [26] developed a Quantum Benders Decomposition Algorithm, transforming integer programming subproblems into Quadratic Unconstrained Binary Optimization (QUBO) models, which were then efficiently solved via Quantum Annealing. A hybrid quantum-classical Column Generation Algorithm was introduced by [27], where the pricing subproblem was optimized using qubit interactions. Another breakthrough in deterministic quantum optimization was the development of a Quantum Interior Point Method by [28], who demonstrated that leveraging quantum solvers in the Newton step of the algorithm can significantly accelerate convergence for large-scale linear and semidefinite programming problems.

One of the most promising applications of quantum optimization algorithms is in Vehicle Routing Problems, which are closely related to Dynamic Ridesharing Problems. Early work in this field was conducted by [29], who applied Quantum Annealing to solve large-scale VRP instances, achieving superior convergence speed and solution quality compared to classical methods. Building upon this, [30] extended the quantum approach to Truck Routing with Time Windows, developing a Quantum Evolutionary Algorithm combined with a Greedy Local Search heuristic to escape local optima. Later, [31] and [32] formulated Capacitated VRP with and without time windows as a QUBO model, solving it via Quantum Annealing. More recent work by [33] focused on heterogeneous vehicle fleets, where a Quantum Approximate Optimization Algorithm (QAOA) was employed to verify whether different quantum variable assignments formed a valid Hamiltonian tour while satisfying vehicle capacity constraints. A similar quantum-inspired approach was adopted by [34] for network routing problems.

The significance of this study can be summarized in two key aspects, though its value extends beyond these points and is not limited to the Dynamic Ridesharing Problem alone. These aspects include modeling the Dynamic Ridesharing Problem using quantum computing and introducing a framework for implementing quantum solution methodologies. This paper is structured into five chapters. Chapter 2 provides essential background on quantum computing to help readers better understand the problem. Chapter 3 first presents the classical model of the Dynamic Ridesharing Problem and then builds upon the concepts introduced in Chapter 2 to develop a quantum-based model. Chapter 4 explores how this model can be applied and discusses the implementation of quantum-inspired solution methodologies as an alternative to classical approaches. Finally, Chapter 5 concludes the paper.

2. Prerequisites for Understanding Quantum Computing

To grasp the concept of quantum mathematical models and explore quantum-inspired solution algorithms in future work, it's crucial to first gain a deeper understanding of quantum computing.

2.1 One qubit

A classical bit has a definite state of either zero or one. However, a quantum bit, which is the fundamental unit of quantum information and is called a qubit, is represented as unit vectors in the complex number plane. Since there are infinitely many unit vectors in the complex number plane, a qubit can have infinitely many possible values. In Figure 1, a schematic representation of a classical bit and a qubit is shown. The notation $|0\rangle$ represents the column vector $\begin{bmatrix}1\\0\end{bmatrix}$, and $|1\rangle$ represents the column vector $\begin{bmatrix}0\\1\end{bmatrix}$. Additionally, $|i\rangle$ represents the column vector $\frac{1}{\sqrt{2}}\begin{bmatrix}1\\i\end{bmatrix}$ and $|-i\rangle$ represents the column vector $\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-i\end{bmatrix}$. Finally, $|+\rangle$ represents the column vector $\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$, and $|-\rangle$ represents the column vector $\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$.

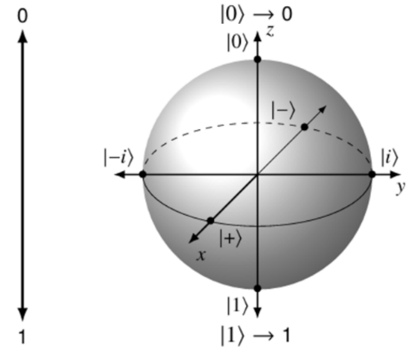

Figure 1. Difference between bit and qubit representation [1]

The symbol represents the state of a qubit ψ and is written as: $|\psi\rangle = a|0\rangle + b|1\rangle$, where $|0\rangle$ and $|1\rangle$ are the unit vectors e_1 and e_2 in the complex number plane. The coefficients a and b are called probability amplitudes and satisfy the relation: $|a|^2 + |b|^2 = 1$. After passing through a computational gate, which we will discuss in more detail later, it is determined whether the qubit takes the value $|0\rangle$ or $|1\rangle$. As a result, $|a|^2$ can be interpreted as the probability of the qubit being measured in the $|0\rangle$ state, and $|b|^2$ as the probability of being measured in the $|1\rangle$ state after computation.

Complex numbers and vectors can be represented in polar form by considering their relative phase differences in magnitude and direction. In other words, as shown in Figure 2, which is known as the Bloch sphere, a qubit can be expressed using the following equation:

$$|\psi\rangle = a|0\rangle + b|1\rangle = r_1 e^{\phi_1 i}|0\rangle + r_2 e^{\phi_2 i}|1\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)e^{\phi i}|1\rangle \quad (1)$$

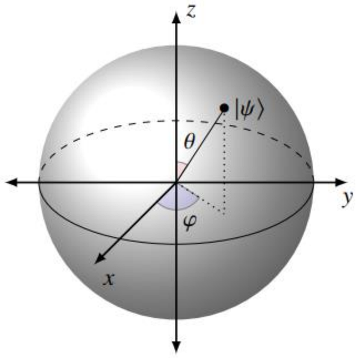

Figure 2. Polar representation of a qubit [1]

2.2 many qubits

Next, we describe the interaction between two qubits. Naturally, the rules discussed here can be extended to more than two qubits. Unlike classical bits, which are completely independent and do not influence each other, qubits are not necessarily independent and can interact with one another. As we have seen, the state of a single qubit can be represented by unit vectors in $\mathbb{C}^2$. Now, if we have two qubits in quantum states $|\psi_1\rangle$ and $|\psi_2\rangle$, where (a_1, b_1) are the probability amplitudes of the first qubit and (a_2, b_2) are the probability amplitudes of the second qubit, then the combined state of these two qubits is represented by their tensor product in $\mathbb{C}^4$, as given in Equation (2). For simplicity and ease of notation, this expression is often rewritten as in Equation (3). If the reader is unfamiliar with the tensor product of vectors in these equations, they can refer to Appendix for more details. In Equation (3), the coefficients a_1a_2, etc., represent the probability amplitudes of each vector. In other words, for example, the coefficient $|a_1a_2|^2$ represents the probability that, after measurement, both qubits take the value $|0\rangle$.

$$|\psi_1\rangle \otimes |\psi_2\rangle = a_1a_2|0\rangle \otimes |0\rangle + a_1b_2|0\rangle \otimes |1\rangle + b_1a_2|1\rangle \otimes |0\rangle + b_1b_2|1\rangle \otimes |1\rangle \tag{2}$$

$$|\psi_1\rangle \otimes |\psi_2\rangle = a_1a_2|00\rangle + a_1b_2|01\rangle + b_1a_2|10\rangle + b_1b_2|11\rangle \tag{3}$$

2.3 Entangled qubits

Qubits that are entangled before measurement are so strongly correlated that measuring one immediately determines the state of the other. This phenomenon does not occur with classical bits and is one of the fundamental differences between quantum and classical computing. The question arises: how can we determine whether two qubits are entangled? To answer this, let Ψ represent the quantum state of two qubits in $\mathbb{C}^4$ The two qubits are **entangled** if and only if their state **cannot** be written as the tensor product of two separate qubit states. In other words, it should not be possible to express Ψ in the form given by Equation (4).

$$\Psi = |\psi_1\rangle \otimes |\psi_2\rangle = (a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) \tag{4}$$

To better understand this concept, let's consider an example. Suppose that Ψ is not an entangled two-qubit state. Also, let $\Psi = 0|00\rangle + \frac{\sqrt{2}}{2}|01\rangle + \frac{\sqrt{2}}{2}|10\rangle + 0|11\rangle$. According to the explanation in Equation (4), there must exist values for a_1, a_2, b_1, and b_2 such that the equation (5) holds. Since there are no values for a_1, a_2, b_1, and b_2 satisfy Equation (5), it is not possible to express Ψ as the tensor product of two separate qubits. Therefore, Ψ represents an entangled two-qubit quantum state.

$$a_1a_2 = 0, \quad a_1b_2 = \frac{\sqrt{2}}{2}, \quad b_1a_2 = \frac{\sqrt{2}}{2}, \quad b_1b_2 = 0 \tag{5}$$

2.4 Amplification of probability amplitudes

By combining a set of single-qubit gates, as discussed in previous sections, along with multi-qubit gates—two of which were examined in this section—and utilizing certain techniques beyond the scope of this thesis, quantum circuits are constructed. These circuits are then used to implement quantum algorithms. However, the main question is: given that qubits exhibit probabilistic and non-deterministic behavior, how can we ensure that the quantum state output of a circuit aligns with our desired outcome? The answer lies in amplifying probability amplitudes. Suppose we have three qubits initially in the $|0\rangle$ state and apply a three-qubit Hadamard gate ($H^{\otimes 3}$) to them. This operation results in an output that is a linear combination of all possible states, as expressed in Equation (6).

$$|\phi\rangle = H^{\otimes 3}|000\rangle = \sum_{j=0}^{7} \frac{1}{\sqrt{8}}|j\rangle_3 = \frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle) \tag{6}$$

Through a process known as probability amplitude amplification, we can increase the likelihood that, upon measurement, the output corresponds to a specific desired state. This is achieved using operators called phase shift and inversion about the mean, which are fundamental techniques in quantum algorithms. For a more detailed study of these techniques, refer to [1]. Of course, this approach is not without challenges. It introduces precision issues in quantum algorithms that must be carefully controlled to ensure accurate results.

2.5 Classical and quantum algorithms

In classical computing, algorithms are defined using the concept of a Turing machine. An algorithm is a finite set of well-defined instructions that guide the execution of operations to solve a problem. If the input string length is n, the computational complexity of an algorithm is typically expressed using Big-O notation $O(\cdot)$. For example, bubble sort has a complexity of $O(n^2)$, while merge sort operates in $O(n \log n)$.

Problems are classified based on whether they can be solved by deterministic or non-deterministic Turing machines. If an algorithm always produces the same output for a given input, it is deterministic. Otherwise, it is non-deterministic, meaning that its output may vary. Key problem classifications include:

- P (Polynomial time): Problems solvable in polynomial time by a deterministic Turing machine.
- NP (Non-deterministic Polynomial time): Problems whose solutions can be verified in polynomial time but may not be solvable in polynomial time.
- NP-hard: Problems at least as hard as the hardest problems in NP.
- NP-complete: Problems that are both NP-hard and belong to NP, meaning they can be verified in polynomial time.

Quantum algorithms, like classical ones, consist of a sequence of instructions executed on a quantum computing machine. However, these instructions must align with quantum mechanics principles, requiring transformations that leverage qubits and quantum gates rather than classical binary logic. Most quantum computing researchers believe that $P \neq NP$. They also suspect that certain NP problems are not in P, but quantum computers can solve them in polynomial time—problems classical computers may never solve efficiently. However, proving this remains an open challenge in theoretical computer science. There are two approaches to comparing quantum and classical algorithms:

1. Theoretical Approach: Reformulating complexity analysis to better capture quantum advantages.
2. Practical Approach: Designing quantum algorithms that solve real-world problems in polynomial time, even when classical algorithms are suspected to lack efficient solutions.

A notable example is Shor's algorithm, which efficiently factors large numbers in polynomial time. While classical algorithms for integer factorization are believed to require super polynomial time, no formal proof of this exists yet [35]. A key metric for comparing quantum and classical algorithms is query complexity—the number of times an algorithm queries its most computationally expensive function. In quantum computing, this measure helps analyze the efficiency of quantum algorithms compared to their classical counterparts.

3. Classical and Quantum Mathematical Modelling

In this section, we first present the classical modeling of the dynamic ride-sharing problem. Then, we attempt to re-model this problem using quantum variables. The quantum modeling in this section will serve as the foundation for quantum solution methods in future works.

3.1 Classical model

In this section, the classical modeling of the dynamic ride-sharing problem is presented. The sets, parameters, and decision variables of the problem are defined, and the objective function, along with the constraints, is formulated. In this problem, it is assumed that each user (passenger or driver) submits their request to the system, including their origin, destination, and preferred departure and arrival times. The transportation network is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting of a set of nodes $(\mathcal{V})$ and edges $\left((i,j) \in \mathcal{E}\right)$ connecting them. The travel costs associated with each route, denoted by t_{ij}, are updated based on real-time traffic data. Additionally, each driver d has a specific capacity c_d for carrying passengers.

Passengers can only be assigned to a single vehicle, and all routes are modeled discretely. The objective of the problem is to optimally assign passengers to drivers in a way that minimizes the total cost arising from deviations in departure and arrival times from users' preferred schedules. Furthermore, it is assumed that traffic constraints and vehicle capacity limitations are satisfied, and the flow of routes within the network is maintained. The notation used in the classical modeling of the problem is detailed in Table 1.

Table 1. Problem's notations

Sets	
D	The set of all drivers
R	The set of all riders
U	The set of all users in the system ($U = D \cup R$)
i, j	The set of graph vertices representing user pickup locations and destinations
F_i^*	The set that includes all nodes j such that the edge $(i, j) \in \mathcal{E}$ exists.
B_i^*	The set that includes all nodes j such that the edge $(j, i) \in \mathcal{E}$ exists.
ori_u	The origin vertex for user $u \in U$
des_u	The destination vertex for user $u \in U$
Parameters	
S_u	Travel preferred start time for user $u \in U$
E_u	Travel preferred end time for user $u \in U$
a_u	Earliness/Tardiness cost coefficient for user $u \in U$ in travel start time
b_u	Earliness/Tardiness cost coefficient for user $u \in U$ in travel end time
h_u	Acceptable tolerance in time for user $u \in U$
g_d	Capacity of vehicle d
c_{ij}	Travel cost from vertex i to vertex j ($(i, j) \in \mathcal{E}$)
t_{ij}	Travel time from vertex i to vertex j ($(i, j) \in \mathcal{E}$) updated by traffic data
Decision Variables	
σ_u	Actual travel start time for user $u \in U$
τ_u	Actual travel end time for user $u \in U$
$y_{d,r}$	It equals 1 if the passenger $r \in R$ is assigned to the driver $d \in D$, and 0 otherwise
$x_{i,j}^u$	It equals 1 if the user $u \in U$ travels through edge $(i, j) \in \mathcal{E}$ along their route, and 0 otherwise

$$\min \sum_{u \in U} f_u(\sigma_u) + g_u(\tau_u) + \sum_{u \in U} \sum_{(i,j) \in \mathcal{E}} c_{ij} \cdot x_{ij}^u \tag{7}$$

Where:

$$f_u(\sigma_u) = \begin{cases} a_u(\sigma_u - S_u), & \sigma_u > S_u + h_u \\ 0, & S_u - h_u \le \sigma_u \le S_u + h_u \\ a_u(S_u - \sigma_u), & \sigma_u < S_u - h_u \end{cases} \tag{8}$$

$$g_u(\tau_u) = \begin{cases} b_u(\tau_u - E_u), & \tau_u > E_u + h_u \\ 0, & E_u - h_u \le \tau_u \le E_u + h_u \\ b_u(E_u - \tau_u), & \tau_u < E_u - h_u \end{cases} \tag{9}$$

Subject to:

$$\tau_u = \sigma_u + \sum_{(i,j) \in \mathcal{E}} t_{ij} \cdot x_{ij}^u; \qquad \forall u \in U \tag{10}$$

$$\sum_{d \in D} y_{d,r} = 1; \qquad \forall r \in R \tag{11}$$

$$\sum_{r\in R} y_{d,r} \le g_d; \qquad \forall d \in D \tag{12}$$

$$\begin{cases} y_{d,r} + x^r_{i,j} - 1 \le x^d_{i,j} \\ y_{d,r} + x^d_{i,j} \ne 2 + 2x^r_{i,j} \end{cases}; \qquad \forall d \in D, \quad \forall r \in R, \quad \forall\, i,j \in \mathcal{V} \tag{13}$$

$$\sum_{j\in F_i^*} x^u_{i,j} - \sum_{j\in B_i^*} x^u_{j,i} = \begin{cases} 1, & if\ i = ori_u \\ -1, & if\ i = des_u\,; \\ 0, & otherwise \end{cases} \qquad \forall u \in U, \quad \forall\, i \in \mathcal{V} \tag{14}$$

$$x^u_{i,j} \in \{0,1\}, \quad y_{d,r} \in \{0,1\}, \quad \sigma_u, \tau_u \ge 0; \quad \forall u \in U, \quad \forall d \in D, \quad \forall r \in R, \quad \forall\, i,j \in \mathcal{V} \tag{15}$$

In the presented modeling, Equation (7) represents the objective function, which minimizes the total travel cost and the deviation from the preferred start and end times for all users in the system. The expanded forms of the functions for start and end times are given in Equations (8) and (9), respectively. Among the constraints of the problem, Equation (10) calculates the arrival time. Equation (11) ensures that each passenger is assigned to only one driver. Equation (12) guarantees that the vehicle capacity constraints for each driver are not violated. Equation (13) ensures that the decision variables $x^r_{i,j}$ and $x^d_{i,j}$ take the same value for edges (i,j) that belong to both the driver's and the passenger's route when they are traveling together. This equation also allows these variables to take different values for drivers and passengers who are not traveling together. Equation (14) maintains the flow in the network. Finally, Equations (15) define the decision variables of the problem.

The problem we are dealing with, if some of its constraints are relaxed, reduces to the VRP. The VRP is classified as an NP-hard problem, as proven in [36]. Since the dynamic ride-sharing problem, as classically modeled, can be reduced to the VRP, we can conclude that it is at least as hard as the VRP and, therefore, belongs to the NP-hard class. This complexity, along with the limitations of heuristic solution methods, motivated us to develop a quantum-based modeling approach and attempt to solve the problem using quantum computing techniques.

3.2 Quantum model

We present the quantum modeling of the problem in a way that allows for the development of hybrid solution methods combining classical and quantum computing. The purpose of this quantum modeling will become clear in the discussion section. To formulate the problem purely with quantum variables, we disregard positive variables. Specifically, we assume that the numerical values of vectors σ and τ are known and denote them by $\bar{\sigma}$ and $\bar{\tau}$. Additionally, for simplicity in representing relationships, we define the notation ξ^u_i to represent the right-hand side of Equation (14).

In the two-dimensional space of complex numbers, we define the vectors $|x^u_{i,j}\rangle = \alpha^u_{i,j}|0\rangle + \beta^u_{i,j}|1\rangle$ and $|y_{d,r}\rangle = \gamma_{d,r}|0\rangle + \delta_{d,r}|1\rangle$ Here, the coefficients of these unit vectors represent the probability amplitudes of the defined quantum states, as explained in section 2. With these definitions, we redefine the problem formulation as follows.

$$\min \sum_{u\in U} f_u(\overline{\sigma_u}) + g_u(\overline{\tau_u}) + \sum_{u\in U}\sum_{(i,j)\in\mathcal{E}} c_{ij}\cdot \langle x^u_{i,j}|1\rangle\langle 1|x^u_{i,j}\rangle \tag{16}$$

Subject to:

$$\overline{\tau_u} = \overline{\sigma_u} + \sum_{(i,j)\in\mathcal{E}} t_{ij}\cdot \langle 1|x^u_{i,j}\rangle\langle x^u_{i,j}|1\rangle\,; \qquad \forall u \in U \tag{17}$$

$$\sum_{d\in D} \langle 1|y_{d,r}\rangle\langle y_{d,r}|1\rangle = 1; \qquad \forall r \in R \tag{18}$$

$$\sum_{r\in R} \langle 1|y_{d,r}\rangle\langle y_{d,r}|1\rangle \le g_d; \qquad \forall d \in D \tag{19}$$

$$\langle 1|y_{d,r}\rangle\langle y_{d,r}|1\rangle + \langle 1|x^r_{i,j}\rangle\langle x^r_{i,j}|1\rangle - 1 \le \langle 1|x^d_{i,j}\rangle\langle x^d_{i,j}|1\rangle; \qquad \forall d \in D, \quad \forall r \in R, \quad \forall\, i,j \in \mathcal{V} \tag{20}$$

$$\langle 1|y_{d,r}\rangle\langle y_{d,r}|1\rangle + \langle 1|x^d_{i,j}\rangle\langle x^d_{i,j}|1\rangle \ne 2 + 2\langle 1|x^r_{i,j}\rangle\langle x^r_{i,j}|1\rangle \tag{21}$$

$$\sum_{j\in F_i^*}\langle 1|x_{i,j}^u\rangle\langle x_{i,j}^u|1\rangle-\sum_{j\in B_i^*}\langle 1|x_{j,i}^u\rangle\langle x_{j,i}^u|1\rangle=\begin{cases}1, & if\ i=ori_u\\ -1, & if\ i=des_u\,;\\ 0, & otherwise\end{cases}\qquad \forall u\in U,\quad \forall\, i\in\mathcal{V} \tag{22}$$

$$|x_{i,j}^u\rangle, |y_{d,r}\rangle\in\mathbb{C}^2;\quad \forall u\in U,\quad \forall d\in D,\quad \forall r\in R,\quad \forall\, i,j\in\mathcal{V} \tag{23}$$

In the above modeling approach, it is assumed that the values of non-negative variables are known, and this model is merely a transformation of the binary model projected into the complex number space. The purpose of this quantum formulation, which represents only the binary aspect of the original model, is to enable the decomposition of the binary submodel and solve it using quantum algorithms. The quantum formulation of the original model transforms classical binary decision variables into quantum states using Dirac notation. In the classical model, the variables $x_{i,j}^u$ and $y_{d,r}$ can only take values of 0 or 1. However, in the quantum model, these variables are defined as superpositions in the basis states $|0\rangle$ and $|1\rangle$.

The probability of measuring each variable in state $|1\rangle$ is given by $|\beta_{i,j}^u|^2$ and $|\delta_{d,r}|^2$, respectively. These probabilities serve as the expected equivalent values of the classical binary variables. This transformation allows the optimization problem to retain its original combinatorial structure while achieving a continuous relaxation. It enables the application of quantum-inspired optimization methods such as the Quantum QAOA and Quantum Annealing. Consequently, by leveraging Dirac notation and the concept of superposition, the model can harness new computational capabilities, leading to more efficient solutions for complex problems compared to classical models.

4. Discussion

In certain general cases of convex optimization, an algebraic structure known as the Graver basis has been introduced and proven to serve as an effective solution methodology [37]. However, computing this basis on classical computers is computationally expensive. With the advancement of quantum computing and the development of general quantum algorithms, researchers with access to quantum hardware—such as those in [38]—have demonstrated that the Graver basis can be computed significantly faster using quantum annealing. Leveraging this capability, they have developed multiple optimization algorithms, including GAMA, to solve specific instances of QUBO.

Given that we have successfully formulated a mathematical model for the dynamic ride-sharing problem based on quantum computing, it is important to ensure that algorithms such as GAMA [38], which operate in QUBO form, can be effectively utilized. In future work, when quantum hardware becomes accessible, our quantum model can be readily transformed into a QUBO representation and solved efficiently using the GAMA algorithm, achieving significantly faster performance compared to classical branch-and-bound methods, as demonstrated in [39]. However, due to the presence of positive variables (σ and τ) in the classical model, integrating the GAMA algorithm with decomposition-based approaches such as Benders Decomposition may provide a more effective solution framework.

5. Conclusion

In this paper, we first conducted a comprehensive review of the literature on the dynamic ride-sharing problem, a well-known NP-hard problem in optimization, and explored recent advancements in quantum optimization. We then introduced fundamental quantum computing concepts necessary for understanding the subsequent sections. Following this, we presented both the classical and quantum mathematical formulations of the problem. While the objective function is convex, it is also nonlinear, making the development of an exact classical algorithm for solving it particularly challenging.

In the discussion section, we highlighted that with the increasing accessibility of quantum hardware in the coming years, researchers will be able to solve the quantum formulation of this problem using Quantum-Inspired Benders Decomposition. Recent studies have demonstrated that quantum computing can solve similar classes of optimization problems significantly faster and with lower error margins compared to classical methods such as Branch and Bound. This direction presents a promising avenue for future research, particularly when quantum computing resources become more widely available. Furthermore, our proposed approach is not limited to the dynamic ride-sharing problem; it can be extended to a wide range of NP-hard optimization problems, offering broader applicability in the field of quantum optimization.

References

[1] R. S. Sutor. Dancing with Qubits: How quantum computing works and how it can change the world. Packt Publishing Ltd, 2019.
[2] A. Di Febbraro, E. Gattorna, and N. Sacco. Optimization of dynamic ridesharing systems. Transportation research record, 2359(1):44–50, 2013.
[3] H. N. Psaraftis. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transportation Science, 1980.
[4] W. B. Powell. A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers. Transportation Science, 1996.
[5] M. Z. Spivey and W. B. Powell. The dynamic assignment problem. Transportation Science, 2004.
[6] G. Berbeglia, J. Cordeau, and G. Laporte. Dynamic pickup and delivery problems. European Journal of Operational Research, 2010.

[7] M. Schilde, K. F. Doerner, and R. F. Hartl. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Computers Operations Research, 2011.
[8] A. Tafreshian and N. Masoud. Trip-based graph partitioning in dynamic ridesharing. Transportation Research Part C-emerging Technologies, 2020.
[9] A. G. H. Kek, Q. Meng, R. L. Cheu, and C. H. Fung. A decision support system for vehicle relocation operations in carsharing systems. Transportation Research Part E-logistics and Transportation Review, 2009.
[10] A. Najmi, D. Rey, and T. H. Rashidi. Novel dynamic formulations for real-time ride-sharing systems. Transportation Research Part E-logistics and Transportation Review, 2017.
[11] N. Masoud and R. Jayakrishnan. A decomposition algorithm to solve the multi-hop peer-to-peer ride-matching problem. Transportation Research Part B-methodological, 2017.
[12] N. Masoud, R. Lloret-Batlle, and R. Jayakrishnan. Using bilateral trading to increase ridership and user permanence in ridesharing systems. Transportation Research Part E-logistics and Transportation Review, 2017.
[13] N. Masoud and R. Jayakrishnan. A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system. Transportation Research Part B-methodological, 2017.
[14] M. Tamannaei and I. Irandoost. Carpooling problem: A new mathematical model, branch-and-bound, and heuristic beam search algorithm. Journal of Intelligent Transportation Systems, 2019.
[15] A. Lee and M. W. P. Savelsbergh. Dynamic ridesharing: Is there a role for dedicated drivers? Transportation Research Part B-methodological, 2015.
[16] M. Stiglic, N. Agatz, M. W. P. Savelsbergh, and M. Gradišar. Making dynamic ride-sharing work: The impact of driver and rider flexibility. Transportation Research Part E-logistics and Transportation Review, 2016.
[17] D. Wang, Q. Wang, Y. Yin, and T. Cheng. Optimization of ride-sharing with passenger transfer via deep reinforcement learning. Transportation Research Part E: Logistics and Transportation Review, 2023.
[18] P. He, J. G. Jin, and F. Schulte. The flexible airport bus and last-mile ride-sharing problem: Math-heuristic and metaheuristic approaches. Transportation Research Part E: Logistics and Transportation Review, 2024.
[19] A. Narayanan and M. Moore. Quantum-inspired genetic algorithms. Proceedings of IEEE International Conference on Evolutionary Computation, 1996.
[20] K. Han, K.-H. Han, and J.-H. Kim. Genetic quantum algorithm and its application to combinatorial optimization problem. Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), 2000.
[21] H. Teng, B. Yang, and B. Zhao. A new mutative scale chaos optimization quantum genetic algorithm. Chinese Control and Decision Conference, 2008.
[22] S. Zhao, G. Xu, T. Tao, and L. Liang. Real-coded chaotic quantum-inspired genetic algorithm for training of fuzzy neural networks. Computers Mathematics With Applications, 2009.
[23] J. Sun, B. Feng, and W. Xu. Particle swarm optimization with particles having quantum behavior. Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), 2004.
[24] J. Liu, W. Xu, and J. Sun. Quantum-behaved particle swarm optimization with mutation operator. IEEE International Conference on Tools with Artificial Intelligence, 2005.
[25] G. Nannicini. Fast quantum subroutines for the simplex method. Operations Research, 72(2):763–780, 2024.
[26] Z. Zhao, L. Fan, and Z. Han. Hybrid quantum benders' decomposition for mixed-integer linear programming. In 2022 IEEE Wireless Communications and Networking Conference (WCNC), pages 2536–2540. IEEE, 2022.
[27] W. da Silva Coelho, L. Henriet, and L.-P. Henry. Quantum pricing-based column-generation framework for hard combinatorial problems. Physical Review A, 107(3):032426, 2023.
[28] I. Kerenidis and A. Prakash. A quantum interior point method for lps and sdps. ACM Transactions on Quantum Computing, 2020.
[29] A. Syrichas and A. Crispin. Large-scale vehicle routing problems: Quantum annealing, tunings and results. Computers Operations Research, 2017.
[30] Z. Wang and J. Dai. Separated vehicle scheduling optimisation for container trucking transportation based on hybrid quantum evolutionary algorithm. International Journal of Computing Science and Mathematics, 2017.
[31] H. Irie, G. Wongpaisarnsin, M. Terabe, A. Miki, and S. Taguchi. Quantum annealing of vehicle routing problem with time, state and capacity. QTOP@NetSys, 2019.
[32] S. M. Harwood, C. Gambella, D. Trenev, A. Simonetto, D. E. Bernal, and D. Greenberg. Formulating and solving routing problems on quantum computers. IEEE Transactions on Quantum Engineering, 2021.
[33] D. Fitzek, T. Ghandriz, L. Laine, M. Granath, and A. F. Kockum. Applying quantum approximate optimization to the heterogeneous vehicle routing problem. Scientific Reports, 2021.
[34] S. Chiew, K. Poirier, R. Mishra, U. Bornheimer, E. Munro, S. H. Foon, C. W. Chen, W. Lim, and C. Nga. Multiobjective optimization and network routing with near-term quantum computers. IEEE Transactions on Quantum Engineering, 2023.
[35] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review, 41(2):303–332, 1999.
[36] R. M. Karp. Reducibility among combinatorial problems. Springer, 2010.
[37] J. A. De Loera, R. Hemmecke, S. Onn, U. Rothblum, and R. Weismantel. Convex integer maximization via graver bases. Journal of Pure and Applied Algebra, 213(8):1569–1577, 2009.
[38] H. Alghassi, R. Dridi, and S. Tayur. Graver bases via quantum annealing with application to non-linear integer programs. arXiv preprint arXiv:1902.04215, 2019.
[39] A. Karahalios, S. Tayur, A. Tenneti, A. Pashapour, F. S. Salman, and B. Yıldız. A quantum-inspired bilevel optimization algorithm for the first responder network design problem. INFORMS Journal on Computing, 37(1):172–188, 2025.