Journal of Industrial and Systems Engineering
Vol. 17, Special issue: 20th Iranian International Industrial Engineering Conference
Autumn (October) 2025; pp. 81-92

JISE

A Novel Hybrid Quantum-Enhanced Approach to Solve the Complex Last-Mile Delivery Problem with Mobile Hubs, Roaming Delivery, and Crowd Shippers

Sadjad Khalesi[1], Mohammad Reza Akbari Jokar[1*], Moslem Habibi[1]
[1]*Industrial engineering department, Sharif university of technology, Tehran, Iran*

Abstract
This paper presents the Hybrid Quantum-Enhanced Simulated Annealer (QSA), a novel quantum-classical algorithm for optimizing complex last-mile delivery problems. Unlike traditional search-based methods like Variable Neighborhood Search (VNS) and Adaptive Large Neighborhood Search (ALNS), QSA integrates quantum exploration with classical optimization to enhance solution quality and efficiency. Computational experiments using the Qiskit quantum simulator (up to 20 qubits) and IBM Quantum Platform (up to 50 qubits) show that QSA outperforms VNS and ALNS, achieving near-optimal solutions in just 3 iterations versus 500 for traditional methods. This highlights QSA's potential for large-scale combinatorial optimization. While quantum hardware remains limited, the study demonstrates the transformative promise of quantum computing in logistics, paving the way for future advancements in hybrid methodologies.

Keywords: Complex Urban logistics; Quantum Simulated Annealing; Local Search Algorithm; IBM Quantum Platform; Qiskit Quantum Simulator.

1. Introduction

As a generalization of the Traveling Salesman Problem (TSP), the Vehicle Routing Problem (VRP) seeks to determine the most efficient routes for a fleet of vehicles to deliver goods or services to a set of customers, subject to constraints such as vehicle capacity, time windows, and route feasibility. The problem's complexity grows exponentially with the number of customers and vehicles, making it a prime candidate for advanced computational techniques. Classical approaches, including heuristic and metaheuristic algorithms, have been widely employed to tackle the VRP, but their scalability and efficiency are often limited by the problem's NP-hard nature. In this context, quantum computing has emerged as a promising paradigm, offering the potential to solve complex optimization problems exponentially faster than classical methods. Quantum computing leverages the principles of quantum mechanics, such as superposition, entanglement, and quantum tunneling, to perform computations that are infeasible for classical computers. Recent breakthroughs, such as the demonstration of quantum supremacy [1], have underscored the transformative potential of quantum technologies. Quantum supremacy, achieved using a 53-qubit superconducting processor, demonstrated that a quantum computer could solve a specific problem in 200 seconds that would otherwise take a classical supercomputer 10,000 years. This milestone has catalyzed interest in applying quantum computing to real-world optimization challenges. In the context of combinatorial optimization, quantum annealing and gate-based quantum algorithms have shown particular promise for tackling NP-hard problems like the VRP ([2];[3]). These quantum approaches offer the possibility of exploring vast solution spaces more efficiently, potentially yielding optimal or near-optimal solutions in significantly reduced timeframes. The potential impact of quantum computing on the VRP extends beyond mere computational speedups. By enabling the exploration of previously intractable solution spaces, quantum algorithms could uncover novel routing strategies that significantly reduce costs, improve service levels, and enhance sustainability. This research aims to contribute to the growing body of knowledge on quantum-based algorithms for the VRP by addressing key challenges and exploring new methodologies. Specifically, the study focuses on the development of hybrid quantum-classical algorithms that can efficiently solve large-scale VRP instances with complex constraints. By leveraging advancements in quantum hardware and algorithmic techniques, the research seeks to bridge

* Corresponding Author ISSN: 1735-8272,

the gap between theoretical potential and practical applicability. The main contributions of this research can be summarized as follows:

- This study extends the application of quantum computing to logistics by introducing Quantum based Simulated Annealing (QSA), a hybrid quantum-classical algorithm tailored for multi-echelon last-mile delivery problems, including static/mobile hubs, roaming delivery locations, and crowd-shipping via occasional drivers.
- We propose a novel integration of quantum exploration with classical optimization techniques, enabling QSA to efficiently navigate complex solution spaces and achieve near-optimal solutions in just 3 iterations, compared to 500 iterations required by traditional methods like VNS and ALNS.
- This research provides critical insights for logistics and supply chain management by demonstrating the superior performance of QSA in terms of solution quality and computational efficiency, validated on both Qiskit (up to 20 qubits) and IBM Quantum Platform (up to 50 qubits).
- Advancing toward scalable quantum solutions, QSA highlights the potential of hybrid quantum-classical methodologies to address large-scale combinatorial optimization problems, even with current hardware limitations.
- By bridging the gap between quantum computing and real-world logistics challenges, this study paves the way for future advancements in quantum-enabled optimization, offering transformative potential for the logistics industry.

The remainder of the paper is structured as follows: Section 2 reviews related works. Section 3 outlines problem assumptions and a mathematical formulation for a crowd-based multi-echelon last-mile delivery problem with roaming locations. Section 4 describes proposed algorithms, while Section 5 presents computational results. Section 6 concludes with future research directions.

2. Literature Review

The application of quantum-based algorithms to solve combinatorial optimization problems, particularly the VRP, has garnered significant attention in recent years. A significant portion of the literature focuses on quantum annealing, a quantum computing paradigm particularly suited for combinatorial optimization. [4] applied quantum annealing to the dynamic multi-depot capacitated VRP, demonstrating its efficiency in handling logistics optimization. [5] introduced hybrid quantum annealing algorithms, such as the DBSCAN Solver and Solution Partitioning Solver, which outperformed classical methods in solving VRP and its variants. These studies highlight the adaptability of quantum annealing to real-world constraints, such as heterogeneous fleets and priority deliveries, as further demonstrated by [6] in their quantum-classical hybrid approach, Q4RPD, for package delivery routing. Hybrid quantum-classical approaches have emerged as a practical strategy to overcome the limitations of near-term quantum devices. Feld et al. (2019) [7] proposed a hybrid method for solving the Capacitated VRP (CVRP), combining quantum annealing with classical heuristics to achieve feasible solutions. [8] extended this approach to supply chain logistics, demonstrating the potential of hybrid methods to tackle large-scale problems by breaking them into smaller, manageable instances. These studies highlight the synergy between classical and quantum computing. In addition to quantum annealing, gate-based quantum computing has shown promise for solving routing problems. [9] explored the application of quantum algorithms to solve the TSP using IBM's quantum computing platform, highlighting the potential of gate-based quantum computing in addressing NP-hard problems. Their work demonstrated how quantum algorithms could be implemented on existing quantum hardware, paving the way for future research in this area.

Despite these advancements, challenges remain in scaling quantum algorithms to practically relevant problem sizes. [10] discussed the heuristic nature of most quantum optimization algorithms and the lack of rigorous guarantees, emphasizing the need for further research to enhance their scalability and robustness. [11] and [12] explored techniques such as circuit cutting and randomized measurements to simulate large quantum circuits on smaller devices, offering promising directions for overcoming hardware limitations. [13] evaluated the performance of commercial quantum annealing solvers for the CVRP, finding that solution quality degraded as problem complexity increased. Their study emphasized the importance of minimizing constraint density in problem formulations to improve the performance of quantum solvers. Similarly, [14] explored the combination of graph shrinking and circuit cutting to reduce the size of quantum circuits, making them more feasible for solving combinatorial optimization problems like the CVRP. Their work demonstrated the effectiveness of these methods in retrieving optimal solutions for the TSP using smaller quantum circuits. The integration of quantum computing with other computational paradigms, such as reinforcement learning, has also been explored. [15] demonstrated that shallow quantum circuits could replace classical attention head layers in reinforcement learning policies for VRP, highlighting the feasibility of hybrid classical-quantum approaches. The application of quantum computing to real-world transportation problems has also been a focus of recent studies. [16] introduced a QUBO model for the Transport Network Design Problem (TNDP), demonstrating the computational benefits of quantum annealing over traditional meta-heuristic methods. Their research highlighted the potential of quantum computing in solving large-scale transportation optimization problems efficiently. Similarly, [17] explored the

use of quantum annealing for traffic flow optimization, demonstrating how quantum computing could improve network utilization and reduce congestion.

In conclusion, quantum-based algorithms offer significant potential for solving the VRP and its variants, leveraging the computational advantages of quantum computing to address the limitations of classical methods. Quantum annealing, hybrid quantum-classical approaches, and QUBO formulations have emerged as key methodologies in this field, demonstrating promising results in both theoretical and practical applications.

3. Model Description

Let G= (V, A) denote a complete directed graph with node set V and arc set A, where node set V is a combination of the main depot, satellites and customers' locations that is shown in equation 1. The set $A_1 = A = \{(i,j)|i,j \in V, i \neq j\}$ denotes the set of arcs in first echelon and the set A_2 is considered for second echelon arcs and determined by equations 2-4.

$$V = \{0\} \cup V^s \cup V^c = \{depot\ location\} \cup \{satellites' locations\} \cup \{customers'\ locations\} \quad (1)$$

$$A_2 = A_3 \backslash A_4 \quad (2)$$

$$A_3 = \{(i,j) \in A | i,j \in V \backslash \{0\}, i \neq j\} \quad (3)$$

$$A_4 = \{(i,j) \in A | i,j \in V^s, i \neq j\} \quad (4)$$

To formulate the Crowd-based multi echelon vehicle routing problem with roaming delivery locations and mobile satellites (MEVRPRDL-MS), the following notations have been used:

Table 1.List of indices

indices :	
i: $index\ of\ all\ nodes$	V^s: $satellites'\ locations\ set$
j: $index\ of\ first\ echelon\ vehicles$	V^c: $customers' locations\ set$
l: $index\ of\ crowd\ shippers$	c': $customers' index$
Cs: $crowd\ shippers\ set$	c: $customers' locations\ index$
s: $index\ of\ sattelites$	

Table 2. List of parameters

Parameters
$\|S\|$: $number\ of\ sattelite$
$\|C\|$: $number\ of\ customers$
M^{fev}: $capacity\ of\ fev\ vehicle$
M^{cs_l}: $capacity\ of\ cs\ vehicle\ l$
T: $time\ horizon\ of\ planning\ period$
$GP_{c'}$: $geographic\ profile\ of\ customer\ c';\ GP_{c'} \subseteq V^c; \|GP_{c'}\| = g\ for\ all\ customer\ c'$
$GR_l = (O_x^l, O_y^l, \alpha^l)$: $geographical\ region\ of\ Cs_l\ presence\ to\ serve\ customers$
O_x^l: $longitude\ coordination\ of\ Cs\ l$
O_y^l: $latitude\ coordination\ of\ Cs\ l$
α^l: $Covering\ radius\ of\ Cs\ l$
$long_i$: $longitude\ coordinate\ of\ node\ i \in V^s \cup V^c$
lat_i: $latitude\ coordinate\ of\ node\ i \in V^s \cup V^c$
BM^d: $big\ value\ (m)\ for\ distance$
g: $customer\ geographical\ profile\ size$
p_s: $handling\ cost\ of\ sattelite\ s$
d_i: $demand\ of\ node\ i$
a_i: $lower\ bound\ of\ time\ window\ for\ node\ i \in V^s \cup V^c$
b_i: $upper\ bound\ of\ time\ window\ for\ node\ i \in V^s \cup V^c$
e_l: $lower\ bound\ of\ available\ time\ for\ CS\ l$
f_l: $upper\ bound\ of\ available\ time\ for\ CS\ l$
$c_{i,j}$: $routing\ cost\ from\ i\ to\ j\ by\ first\ echelon\ vehicle(fev)$
$w_{i,j}$: $routing\ cost\ from\ i\ to\ j\ by\ crowd\ shipper(cs)$
$t_{i,j}^1$: $travel\ time\ which\ is\ related\ to\ (i,j) \in A_1\ by\ fev\ at\ first\ echelon$
$t_{i,j,l}^2$: $travel\ time\ which\ is\ related\ to\ (i,j) \in A_2\ by\ crowdshipper\ l\ at\ second\ echelon$

Table 3. List of variables

Variables
$X_{i,j,k}^1 = \begin{cases} 1 & if\ k \in fev\ traverses\ arc\ (i,j) \in A_1 \\ 0 & O.W \end{cases}$
$X_{i,j,s,l}^2 = \begin{cases} 1 & if\ arc\ (i,j) \in A_2\ is\ traversed\ from\ satellite\ s \in V^s\ by\ Cs_l \\ 0 & O.W \end{cases}$
$q_{i,k} \in R^+$: $amount\ of\ parcels\ being\ transported\ from\ depot\ to\ node\ i \in V^s \cup V^c\ by\ vehicle\ k \in fev$
$h_{i,j,s,l} \in R^+$: $load\ of\ Cs_l\ dispatched\ from\ satellite\ s\ uses\ arc(i,j) \in A_2$
$\tau_i^1 \in [0,T]$: $time\ of\ departure\ after\ visit\ node\ i \in V^s \cup V^c\ by\ a\ fev$
$\tau_{i,l}^2 \in [0,T]$: $time\ of\ departure\ after\ visit\ node\ i \in V^c\ by\ crowdshipper\ l\ at\ second\ echelon$
$u_{i,k} \in Z^+$: $auxiliary\ variable\ for\ eliminating\ subtours\ in\ the\ first\ echelon$

The mathematical formulation for Crowd-based MEVRPRDL-MS is then given as follows:

$$Min\ Z = \sum_{k\in fev}\sum_{(i,j)\in A_1} c_{i,j}.X^1_{i,j,k} + \sum_{s\in V^s}\sum_{(i,j)\in A_2}\sum_{l\in Cs} w_{i,j}.X^2_{i,j,s,l} + \sum_{k\in fev}\sum_{s\in V^s} p_s.q_{s,k} \quad (5)$$

$S.t.$

$$\sum_{(i,j)\in A_1} X^1_{i,j,k} = \sum_{(j,i)\in A_1} X^1_{j,i,k} \qquad \forall i \in V^s \cup V^c \cup \{0\}, k \in fev \quad (6)$$

$$\sum_{(i,j)\in A_1} X^1_{i,j,k} \le 1 \qquad \forall i \in V^s \cup V^c \cup \{0\}, k \in fev \quad (7)$$

$$u_{i,k} - u_{j,k} + (|S| + |C|).X^1_{i,j,k} \le (|S| + |C|) - 1 \qquad \forall i,j \in V^s \cup V^c, i \ne j\ , k \in fev \quad (8)$$

$$q_{i,k} \le M^{fev}.\sum_{(j,i)\in A_1} X^1_{j,i,k} \qquad \forall i \in V^s \cup V^c, k \in fev \quad (9)$$

$$\sum_{j\in GP_{c'}}\sum_{i\in V}\sum_{k\in fev} X^1_{i,j,k} \le 1 \qquad \forall c' \in C' \quad (10)$$

$$\sum_{j\in GP_{c'}}\sum_{i\in V^s\cup V^c}\sum_{l\in Cs}\sum_{s\in V^s} X^2_{i,j,s,l} \le 1 \qquad \forall c' \in C' \quad (11)$$

$$\sum_{j\in GP_{c'}}\sum_{i\in V}\sum_{k\in fev} X^1_{i,j,k} + \sum_{j\in GP_{c'}}\sum_{i\in V^s\cup V^c}\sum_{l\in Cs}\sum_{s\in V^s} X^2_{i,j,s,l} = 1 \qquad \forall c' \in C' \quad (12)$$

$$\sum_{i\in V^s\cup V^c} q_{i,k} \le M^{fev} \qquad \forall\ k \in fev \quad (13)$$

$$\sum_{k\in fev} q_{s,k} = \sum_{j\in V^c}\sum_{l\in Cs} h_{s,j,s,l} \qquad \forall\ s \in V^s \quad (14)$$

$$\sum_{k\in fev} q_{c,k} = d_c.\sum_{j\in V}\sum_{k\in fev} X^1_{j,c,k} \qquad \forall\ c \in V^c \quad (15)$$

$$\sum_{(i,j)\in A_2} X^2_{i,j,s,l} = \sum_{(j,i)\in A_2} X^2_{j,i,s,l} \qquad \forall i \in V^s \cup V^c, s \in V^s, l \in Cs \quad (16)$$

$$\sum_{i\in GP_{c'}}\sum_{l\in Cs}\sum_{s\in V^s}\sum_{(j,i)\in A_2} h_{j,i,s,l} - \sum_{i\in GP_{c'}}\sum_{l\in Cs}\sum_{s\in V^s}\sum_{(i,j)\in A_2} h_{i,j,s,l} = {}^1\!/_g \sum_{i\in GP_{c'}} d_i.(1 - \sum_{i\in GP_{c'}}\sum_{j\in V}\sum_{k\in fev} X^1_{j,i,k}) \qquad \forall c' \in C' \quad (17)$$

$$h_{i,j,s,l} \le M^{Cs}_l.X^2_{i,j,s,l} \qquad \forall (i,j) \in A_2, s \in V^s, l \in Cs \quad (18)$$

$$X^2_{i,s,s',l} = 0 \qquad \forall i \in V^s \cup V^c, s, s' \in V^s, s \ne s', l \in Cs \quad (19)$$

$$X^2_{s,j,s',l} = 0 \qquad \forall j \in V^s \cup V^c, s, s' \in V^s, s \ne s', l \in Cs \quad (20)$$

$$X^2_{s,s,s,l} = 0 \qquad \forall s \in V^s, l \in Cs \quad (21)$$

$$\sum_{i\in GP_{c'}}\sum_{j\in GP_{c'}}\sum_{k\in fev} X^1_{i,j,k} = 0 \qquad \forall c' \in C' \quad (22)$$

$$\sum_{i\in GP_{c'}}\sum_{j\in GP_{c'}}\sum_{s\in V^s}\sum_{l\in Cs} X^2_{i,j,s,l} = 0 \qquad \forall c' \in C' \quad (23)$$

$$\sum_{j\in V^s\cup V^c} X^1_{0,j,k} \le 1 \qquad \forall\ k \in fev \quad (24)$$

$$\sum_{s\in V^s}\sum_{j\in V^s\cup V^c} X^2_{s,j,s,l} \le 1 \qquad \forall\ l \in Cs \quad (25)$$

$$\tau^1_i + \sum_{k\in fev} t^1_{i,j}.X^1_{i,j,k} \le \tau^1_j + T.(1 - \sum_{k\in fev} X^1_{i,j,k}) \qquad \forall i \in V^s \cup V^c, j \in V^s \cup V^c \backslash \{i\} \quad (26)$$

$$\sum_{k\in fev} t^1_{0,j}.X^1_{0,j,k} \le \tau^1_j + T.(1 - \sum_{k\in fev} X^1_{0,j,k}) \qquad \forall j \in V^s \cup V^c \quad (27)$$

$$\tau^2_{i,l} + \sum_{s\in V^s} t^2_{i,j,l}.X^2_{i,j,s,l} \le \tau^2_{j,l} + T(1 - \sum_{s\in V^s} X^2_{i,j,s,l}) \qquad \forall i \in V^c, j \in V^c \backslash \{i\}, l \in Cs \quad (28)$$

$$\tau^1_s + t^2_{s,j,l}.X^2_{s,j,s,l} \le \tau^2_{j,l} + T(1 - X^2_{s,j,s,l}) \qquad \forall s \in V^s, j \in V^c, l \in Cs \quad (29)$$

$$a_i.\sum_{j\in V^c\cup V^s\cup\{0\}\backslash\{i\}}\sum_{k\in fev} X^1_{i,j,k} \le \tau^1_i \le b_i.\sum_{j\in V^c\cup V^s\cup\{0\}\backslash\{i\}}\sum_{k\in fev} X^1_{i,j,k} \qquad \forall i \in V^s \cup V^c \quad (30)$$

$$a_i.\sum_{j\in V^c\cup V^s\backslash\{i\}}\sum_{s\in V^s} X^2_{i,j,s,l} \le \tau^2_{i,l} \le b_i.\sum_{j\in V^c\cup V^s\backslash\{i\}}\sum_{s\in V^s} X^2_{i,j,s,l} \qquad \forall i \in V^c, l \in Cs \quad (31)$$

$$\tau^2_{i,l} \ge e_l.X^2_{i,j,s,l} \qquad \forall i \in V^c, j \in V^c \cup V^s, l \in Cs, s \in V^s \quad (32)$$

$$\tau^2_{j,l} \ge \sum_{s\in V^s}\left(e_l + t^2_{s,j,l}\right).X^2_{s,j,s,l} \qquad \forall j \in V^c, l \in Cs \quad (33)$$

$$\tau^2_{j,l} \le f_l \qquad \forall j \in V^c, l \in Cs \quad (34)$$

$$\sqrt{(O^l_x - long_i)^2 - (O^l_y - lat_i)^2} \le \alpha^l + BM^d.(1 - \sum_{j\in V^c\cup V^s}\sum_{s\in V^s} X^2_{i,j,s,l}) \qquad \forall i \in V^c \cup V^s, l \in Cs \quad (35)$$

$$h_{i,j,s,l} \ge 0 \qquad \forall (i,j) \in A_2, s \in V^s \quad (36)$$

$$u_{i,k} \in Z^+ \qquad \forall i \in V^s \cup V^c, k \in fev \quad (37)$$

$$q_{i,k} \ge 0 \qquad \forall i \in V^s \cup V^c, k \in fev \quad (38)$$

$$X^1_{i,j,k} \in \{0,1\} \qquad \forall (i,j) \in A_1, k \in fev \quad (39)$$

$$X^2_{i,j,s,l} \in \{0,1\} \qquad \forall (i,j) \in A_2, s \in V^s, l \in Cs \quad (40)$$

The objective function minimizes the total routing costs (for vehicles in first echelon and crowd shippers in second echelon) and satellite handling cost. Constraints (6) is the flow conservation constraint for first echelon vehicle at each node that ensure vehicles leave node that they enter. Constraints (7) ensure that every node $i \in A_1$ is visited at most once by vehicle $k \in fev$. Constraints (8) is subtours elimination constraints. According to constraints (9), parcels can be transferred to satellite (s) or customer (c) by vehicle $k \in fev$ if this vehicle (k) traverses arc $(i,j) \in A_1$. Constraints (10)-(12) impose that the demand of each customer can be satisfied by at most one of the first echelon vehicle and so unlike satellites, each customer cannot be visited by multiple FEV. They also guarantee that each customers' demand which is not satisfied by FEV, should be served by exactly one crowd shipper and its allocated satellite. Capacity constraints are provided in constraint (13). Constraints (14) ensure that total flow into satellite 's' will be equal to the total flow out of it to all customers. Constraints (15) guarantee that the total demand of customers is satisfied. Constraints (16) impose second echelon flow conservation. Constraints (17) guarantee that the customers' demands which are assigned to second echelon vehicles (CSs) are satisfied and they will prevent subtours in second echelon. Constraints (18) ensure that the capacity of crowd shippers are not exceeded. Constraints (19)-(21) indicate that two different satellites cannot be available in the same second echelon rout. Constraints (22) and (23) ensure that only one location of customer's geographic profile will be chosen for parcel delivery. Constraints (24) guarantee that each first echelon vehicle can be used at most once. Constraints (25) ensure that each crowd shipper can be assigned at most once. Constraints (26) - (29) ensure that the time gap between visiting two consecutive nodes in a rout will be at least as long as the travel time between those two nodes. Constraints (30) and (31) enforce satellite and customers' time windows in first and second echelon. Constraints (32) - (34) enforce crowd shippers' time windows. Constraints (35) are related to geographical region of crowd shippers which are available to serve customers. Constraints (36) - (40) define the domains of the variables.

4. Hybrid Solution Approaches

The solution approaches employed in this study are composed of two main components. The first component involves algorithms designed to generate appropriate initial solutions that, if possible, are feasible and can be obtained within an acceptable time frame. The second component consists of search-based algorithms aimed at improving the initial solution towards a near-optimal one. To this end, two heuristic algorithms are utilized in the first component. The first heuristic algorithm is a hybrid method that combines a node classification approach with a greedy strategy (HA) to construct an appropriate initial solution. The second algorithm in this component is the sweep heuristic algorithm, originally introduced by Gillett and Miller in 1974 [18]. In the second component, three search-based optimization algorithms are employed. The first two are the classical and widely-used ALNS (Adaptive Large Neighborhood Search) and VNS (Variable Neighborhood Search) algorithms, which have been extensively applied in recent years to solve various types of routing problems, yielding highly satisfactory results as reported in the literature. The third algorithm, which constitutes the primary focus of the present study, is a novel search algorithm based on the principles of quantum mechanics, combined with the classical Simulated Annealing algorithm, referred to as the Quantum -Enhanced Simulated Annealing (QSA) algorithm.

4.1. Quantum- Enhanced Approach

The proposed methodology is grounded in a hybrid paradigm that integrates classical simulated annealing (SA) with quantum-enhanced exploration, aiming to enhance the output of Sweep or HA algorithms. Specifically, this algorithm is intended to serve as a replacement for the ALNS and VNS algorithms employed in previous approaches. The algorithm incorporates the following key components. We will investigate the various steps of our QSA approach, along with the associated calculations for each step.

I. Quantum State Preparation

Encodes the classical solution into a quantum state while preparing superposition for exploration.

Qubit Mapping:

Each customer (excluding depot) is mapped to a qubit:

Qubit $i \leftrightarrow$ Customer $(i+1)$; $|0\rangle = Not\ assigned;\quad |1\rangle = Assigned$

Circuit Operation:

Apply X gates to qubits where customers are present in the initial solution.

Ancilla qubits are initialized in superposition with H gates.

Initialize the state using X-gates for customers in the initial solution:

$$|\psi_{init}\rangle = \bigotimes_{i=1}^{n} X^{b_i}|0\rangle^{\otimes n}, \qquad b_i \in \{0,1\} \tag{45}$$

where $b_i = 1$ if customer i is in the initial solution.

Ancilla Superposition:

Ancilla qubits are prepared in a uniform superposition for later entanglement:

$$|\psi_{ancilla}\rangle = \frac{1}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} |k\rangle, \qquad m = num_{ancill} \tag{46}$$

Adaptive Mixer Circuit

Drives quantum exploration using temperature-dependent rotations and entanglement.

Rotation Angle:

Adaptive RY rotations with temperature-dependent scaling:

$$\theta_i = \beta \cdot (1 + e^{-T}), \quad \beta = adaptive\ parameter, \quad T = temperature, \quad i = index\ of\ the\ qubit\ (or\ the\ customer) \tag{47}$$

Higher $T \Rightarrow$ Larger angles $\Rightarrow$ More exploration.

Entanglement:

CZ gates (controlled-Z) between adjacent qubits and CX (controlled-X) + RZ gates between customer and ancilla qubits:

$$U_{mixer} = \prod_{i=1}^{n} RY(\theta_i) \cdot \prod_{i<j} CZ_{i,j} \cdot \prod_{i=1}^{n} CX_{i,ancilla} RZ(\phi_i) \tag{48}$$

Creates correlations between customer assignments.

Enhanced Phase Separator

Encodes problem constraints and costs into quantum phases.

Pairwise Cost Encoding:

CPHASE gates encode travel costs between customers:

$$\phi_{ij} = \gamma_{\text{layer}} \cdot \frac{d_{ij}}{\max(d)} \cdot \text{cost\_scaling}, \quad d_{ij} = \text{distance}(i,j), \left(\gamma_{\text{layer}} \cdot \frac{d_{ij}}{\max(d)} : normalized\ distance\right) \tag{49}$$

γ decays with temperature: $\gamma = \pi \cdot (1 - e^{-1/T})$.

Ancilla-Mediated Interactions:

Additional RZ rotations on ancilla qubits:

$$U_{phase} = \prod_{i<j} CP(\phi_{ij}) \cdot CX_{i,ancilla} RZ(\phi_{ij}/2) CX_{j,ancilla} \tag{50}$$

Amplifies phase differences for low-cost solutions.

Quantum Annealing Loop

Combines mixer and phase separator into a temperature-driven loop.

Number of Layers:

Increases as temperature decreases:

$$num\_layers = max(2, \lfloor 4.\, e^{-T/100} \rfloor) \tag{51}$$

Deeper circuits at lower T refine solutions.

At high temperatures, fewer layers are used (more exploration).

At low temperatures, more layers are used (more exploitation).

Adaptive Parameter Update:

β is adjusted based on success history:

$$\beta_{new} = \begin{cases} \beta(T).(1+\eta) & if\ cost\ improves \\ \beta(T).(1-\eta) & otherwise \end{cases}, \quad \eta = learning\ rate, \quad \beta(T) = \beta_0.\, e^{-1/T} \tag{52}$$

Temperature decay: $T_{new} = T \cdot \alpha$ ($\alpha = \alpha =$ *cooling rate*).

Parameter Scaling:

Layer-specific parameters for fine-grained control:

$$\beta_{\text{layer}} = \beta_{new}.(1 + 0.1.\, layer) \tag{53}$$

$$\gamma_{\text{layer}} = \gamma.(1 - 0.05.\, layer) \tag{54}$$

Measurement & Solution Decoding

Converts quantum measurements into valid VRP solutions.

Bitstring to Solution:

Fixed Endpoints: Enforce predefined start/end customers for intermediate transfer locations.

Greedy Insertion: Assign unplaced customers to minimize insertion cost:

$$Insertion\ Cost = \Delta Distance + \lambda_{cap} \cdot \Delta Capacity + \lambda_{time} \cdot \Delta Time \quad (55)$$

Validity Checks:

Capacity: $\sum demands \leq vehicle\_capacity$

Time Windows: $t_{arrival} \in [t_{earliest}, t_{latest}],\ t_{arrival} = \sum \frac{d_{prev,curr}}{speed} + service\_times$

Dependencies: $t_{customer} \geq t_{dependent\_customer}$

Summary of Parameters

Temperature-Dependent Parameters

The temperature T decreases over time according to a cooling schedule ($T_{new} = T \cdot \alpha$).

This temperature controls:

The rotation angles θ_i in the mixer layer: $\theta_i = \beta \cdot (1 + e^{-T})$.

The phase scaling γ in the phase separator: $\gamma = \pi \cdot (1 - e^{-1/T})$.

The temperature-dependent scaling factor $\beta(T)$ for the rotation angles θ_i in the mixer layer ($\beta(T) = \beta_0 . e^{-1/T}$).

Adaptive Parameters

β_{new} ensures that the system adapts to the problem's structure during the annealing process.

Layer-Specific Parameters

β_{layer} is the layer-specific scaling factor for the rotation angles in the mixer layer (controls the mixer layer for exploration).

γ_{layer} refers to the layer-specific phase scaling factor used in the enhanced phase separator of the quantum circuit (controls the phase separator for exploitation).

The pseudocode presented provides an outline of the main steps involved in the proposed algorithm. The approach begins with quantum circuits generating candidate solutions through parameterized state exploration. Subsequently, classical heuristics are employed to refine these solutions and address any constraints. Finally, adaptive parameters are utilized to balance exploration and exploitation, which is achieved by considering the annealing temperature and the algorithm's historical performance.

Algorithm 1. Quantum-Enhanced Simulated Annealing Approach

Inputs: Distance matrix (distances), Number of vehicles (num_vehicles), Depot (depot), Initial solution (initial_solution),Vehicle capacities (vehicle_capacity), Customer demands (demands), Customer time windows (time_windows), Service times (service_times), Vehicle speed (vehicle_speed), Fixed endpoints (fixed_endpoints), Arrival time dependencies (arrival_time_dependencies), Initial temperature (T), Cooling rate (r),Max iterations (max_iter)
Outputs: Best solution (best_solution), Best cost (best_cost)
Initialization:
Set: current_solution = initial_solution
Calculate: current_cost = calculate_total_cost(current_solution)
Set: best_solution = current_solution
Set: best_cost = current_cost
While iteration <= max_iter:
  Generate candidate solutions using quantum circuits:
  a. Prepare quantum state from current_solution
  b. Apply adaptive mixer and phase separator
  c. Measure quantum states to obtain quantum_solutions
  Refine candidate solutions classically:
  a. Perform local search (e.g., 2-opt, cross-exchange) on quantum_solutions
  b. Apply destroy-and-repair strategies
  Select best candidate solution:
  a. Let neighbor_solution be the best solution from refined quantum_solutions
  b. Calculate neighbor_cost = calculate_total_cost(neighbor_solution)
  Simulated annealing acceptance criteria:
  a. If neighbor_cost < current_cost:
    Accept neighbor_solution (current_solution = neighbor_solution, current_cost = neighbor_cost)
  b. Else:
    Accept with probability exp (-(neighbor_cost - current_cost) / T)
  Update best solution:
  a. If neighbor_cost < best_cost:
    best_solution = neighbor_solution
    best_cost = neighbor_cost
Update temperature:
  a. T = T * r
End While
return best_solution, best_cost

5. Computational Results

This study draws inspiration from the approach proposed by Reyes et al. [19] in order to generate diverse instances. Two categories of instances have been considered: (1) general instances and (2) realistic instances. General instances consist of multiple customers, each with a demand d_c, which can be fulfilled directly from the depot warehouse or via satellites and crowd logistics agents. During each planning horizon (daily working hours), mobile satellites can occupy various geographic locations across different time windows. Notably, stationary satellites are equivalent to mobile satellites that maintain a fixed geographic position throughout all time windows. These instances include up to four time windows, each lasting three hours, with a total planning horizon of 14 hours, inclusive of return time to the depot. On the other hand, in the set of realistic instances, it is assumed that we have three types of customers. The first group consists of customers who are present only at their homes throughout the entire planning horizon. The second group comprises customers who are at home during certain hours, then go to their workplace and return home from there. The third group consists of customers who start at their homes, then go to their workplace, and subsequently move to another location before finally returning home. In these instances, unlike the samples generated in the study by Reyes et al. [19], there are no pre-defined work clusters, and the geographical positions of the customers can encompass any random coordinates. This feature increases the diversity of generated random samples. Furthermore, in this set of instances, the planning horizon is considered to be 14 hours, spanning from 6 a.m. to 8 p.m. The number of time windows in this set is 6, each with a duration of 2 hours. In these instances, individuals who go to their workplace are divided into two categories: full-time workers and part-time workers. Full-time workers start their work at 8 a.m. and continue until 4 p.m. Part-time workers are assumed to start their work at 8 a.m. and continue until 12 p.m. In the production of examples in this section, instead of considering different geographical profiles for each customer, we assume that the service provider divides the entire planning horizon into equal time slots, and customers can choose whether or not they have the possibility of delivery in each time slot and, if possible, what their geographical position will be in that time window. In this way, the geographical profiles and the total available time for delivery for different customers will vary.

5.1. Results and Performance Analysis for Classical Approaches

In this section, we initially consider general instances. Based on this dataset, we examine and study the performance of each of the presented hybrid algorithms. Simultaneously, we compare the quality and accuracy of their responses with the optimal solution obtained from the CPLEX method. The results of these algorithms are presented in Table 4. In this table, the following information is provided for each instance: the number of customers (c), the number of stationary and mobile satellites (SS, MS), the maximum geographic profiles (|g|), the number of owned vehicles (FEV), the number of crowd shippers (cs), the time horizon (TH), the capacities of owned vehicles (Cap) and crowd shippers (CsCap), the number of available delivery locations in different time windows (Loc), and the percentage of available time for delivery (Avg. % time). The objective function of each solution approach is also provided in this table. When calculating the objective function obtained from the CPLEX method, in cases where the optimal solution is not obtained within a maximum time of 1500 seconds, the last feasible solution is reported. To distinguish them from the optimal solutions, the symbol "(o)" is used alongside the optimal solutions. We perform all the tests using a workstation with 12th Gen Intel(R) Core (TM) i7-12700H at 2.30 GHz and 16 GB of RAM. To further analyze the results, we examine the randomly generated instances in the form of realistic datasets, the details of which are presented in Table 5.

Table 4. Algorithm performance results for general instance

Instance	Instance parameters											Exact and Hybrid Algorithms Outputs							
	C	MS	SS	|g|	FEV	CS	TH	Cap	CsCap	Loc	Avg. % time	Obj_{HA}	Obj_{Sweep}	$Obj_{HA\text{-}VNS}$	$Obj_{Sweep\text{-}VNS}$	$Obj_{HA\text{-}ALNS}$	$Obj_{Sweep\text{-}ALNS}$	Obj_{CPLEX}	Obj^{*}
1	5	0	1	4	5	4	14	5	2	19	85.5	114	222	114	168	114	132	114(o)	114
2	5	0	1	4	5	4	14	5	2	18	81	120	186	120	144	120	126	102(o)	120
3	5	0	1	4	5	4	14	5	2	20	100	126	192	126	192	102	192	102(o)	102
4	5	0	2	4	5	4	14	5	2	20	100	156	216	156	180	156	180	114	156
5	5	0	2	4	5	4	14	5	2	19	85.5	168	174	168	156	168	174	132	156
6	5	0	2	4	5	4	14	5	2	17	76.5	78	96	78	72	78	72	72(o)	72
7	10	1	1	4	5	4	14	5	2	40	100	210	288	210	288	198	282	234	198
8	10	1	1	4	5	4	14	5	2	36	81.0	240	312	222	282	210	258	270	210
9	10	1	1	4	5	4	14	5	2	35	78.7	234	330	234	270	210	204	210	204
10	10	1	2	4	5	4	14	5	2	40	100	180	288	150	282	144	282	198	144
11	10	1	2	4	5	4	14	5	2	37	83.2	210	276	162	234	156	174	-[b]	156
12	10	1	2	4	5	4	14	5	2	36	81.0	156	228	156	228	156	228	216	156
13	15	1	2	4	5	4	14	5	2	60	100	246	inf	234	300	228	282	-	228
14	15	1	2	4	5	4	14	5	2	52	78.0	288	444	264	348	240	306	-	240
15	15	1	2	4	5	4	14	5	2	55	82.5	222	408	222	366	222	330	-	222
16	15	2	2	4	5	4	14	5	2	60	100	246	420	234	372	234	360	-	234

[b] CPLEX solver is unable to obtain a feasible solution within 1500 seconds.

Instance	Instance parameters										Avg. % time	Exact and Hybrid Algorithms Outputs							
	C	MS	SS	|g|	FEV	CS	TH	Cap	CsCap	Loc		Obj_{HA}	Obj_{Sweep}	Obj_{HA-VNS}	$Obj_{Sweep-VNS}$	$Obj_{HA-ALNS}$	$Obj_{Sweep-ALNS}$	Obj_{CPLEX}	Obj^*
17	15	2	2	4	5	4	14	5	2	55	82.5	300	402	282	336	234	312	-	234
18	15	2	2	4	5	4	14	5	2	56	84	342	438	330	360	276	318	-	276
19	30	3	3	4	10	4	14	5	2	120	100	660	774	660	654	534	612	-	534
20	30	3	3	4	10	4	14	5	2	115	86.2	618	702	570	690	474	576	-	474
21	30	3	3	4	10	4	14	5	2	108	81	612	684	600	612	522	570	-	522
22	30	3	4	4	10	4	14	5	2	108	81	666	768	642	690	468	564	-	468
23	30	3	4	4	10	4	14	5	2	120	100	576	708	576	690	546	618	-	546
24	30	3	4	4	10	4	14	5	2	112	84	606	726	594	630	510	552	-	510
25	50	4	4	4	20	8	14	8	3	200	100	960	918	900	846	600	726	-	600
26	50	4	4	4	20	8	14	8	3	191	85.9	792	948	750	840	558	720	-	558
27	50	4	4	4	20	8	14	8	3	188	84.6	840	1152	804	918	546	672	-	546
28	50	4	5	4	20	8	14	8	3	189	85	1044	1050	1026	912	660	750	-	660
29	50	4	5	4	20	8	14	8	3	178	80.1	900	1008	864	792	588	636	-	588
30	50	4	5	4	20	8	14	8	3	183	82.3	870	1080	840	918	516	624	-	516
31	80	5	5	4	20	8	14	8	3	320	100	1452	1278	1404	1098	858	1068	-	858
32	80	5	5	4	20	8	14	8	3	300	84.3	1344	1482	1320	1326	870	1074	-	870
33	80	5	5	4	20	8	14	8	3	285	80.1	1278	1554	1248	1302	912	1038	-	912
34	80	5	6	4	20	8	14	8	3	320	100	1308	1302	1236	1176	834	954	-	834
35	80	5	6	4	20	8	14	8	3	305	85.7	1314	1548	1308	1314	738	1128	-	738

(o)-Optimal value of CPLEX solver.

Table 5. Algorithm performance results for realistic instances

Instance	Instance parameters										Avg. % time	Exact and Hybrid Algorithms Outputs						
	C	MS	SS	|g|	FEV	CS	TH	Cap	CsCap	Loc		Obj_{HA}	Obj_{Sweep}	Obj_{HA-VNS}	$Obj_{Sweep-VNS}$	$Obj_{HA-ALNS}$	$Obj_{Sweep-ALNS}$	Obj^*
36	30	5	5	6	10	40	14	10	3	180	100	576	618	576	606	450	480	450
37	30	5	5	6	10	40	14	10	3	170	80.2	570	600	564	552	396	426	396
38	30	5	5	6	10	40	14	10	3	168	79.3	486	840	450	654	402	516	402
39	40	6	6	6	10	40	14	10	3	240	100	666	816	666	678	534	552	534
40	40	6	6	6	10	40	14	10	3	230	81.4	666	828	666	756	498	600	498
41	40	6	6	6	10	40	14	10	3	221	78.3	738	834	720	726	570	708	570
42	50	7	7	6	10	40	14	10	3	300	100	870	966	858	750	708	636	636
43	50	7	7	6	10	40	14	10	3	285	80.7	822	1122	786	948	582	696	582
44	50	7	7	6	10	40	14	10	3	273	77.3	870	1098	870	906	660	714	660
45	60	8	8	6	15	40	14	10	3	360	100	732	978	726	858	648	684	648
46	60	8	8	6	15	40	14	10	3	349	82.4	1170	1248	1152	1086	780	780	780
47	60	8	8	6	15	40	14	10	3	335	79.1	1152	1434	1152	1242	798	930	798
48	80	9	9	6	15	40	14	10	3	480	100	1566	1602	1566	1344	1248	1290	1248
49	80	9	9	6	15	40	14	10	3	474	83.9	1356	1686	1338	1506	1146	1158	1146
50	80	9	9	6	15	40	14	10	3	456	80.7	1464	1602	1452	1452	1050	1140	1050

5.2. Results and Performance Analysis for Quantum- Simulated Annealing Approach

In this section, the performance of classical hybrid algorithms is compared with algorithms enhanced by the hybrid Quantum-Enhanced Simulated Annealer Approach (QSA) using the previously discussed instances. It is important to note that, due to the high computational demands of quantum algorithms, examples involving fewer than 20 qubits were executed on classical computers using the AerSimulator. In contrast, examples requiring more than 20 qubits were implemented on quantum computers via the IBM Qiskit quantum platform. It should also be noted that access to the free-tier IBM quantum platform was limited to a maximum of 600 seconds of runtime. Consequently, it was not feasible to solve all 50 examples presented in earlier sections using quantum computers. As a result, only a subset of these examples, involving up to 50 customers, was addressed in this section. Additionally, to maximize the utility of quantum computers, only three iterations of the quantum algorithms were performed for each problem, and the best output from the third iteration is reported in the table below. Therefore, the comparison presented here is essentially between three iterations of quantum algorithms and 500 iterations of classical algorithms. As such, comparing the solution times between the two approaches is not meaningful, given the significant differences in the number of iterations and the hardware employed. Naturally, the running times on quantum computers are expected to be significantly shorter than those on classical computers.

Table 6. Comparison between Quantum-Simulated Annealing Approach and hybrid algorithms

Instance Number	Obj_{HA}	Obj_{Sweep}	Obj_{HA-VNS}	$Obj_{Sweep-VNS}$	$Obj_{HA-ALNS}$	$Obj_{Sweep-ALNS}$	Obj_{HA-QSA}	$Obj_{Sweep-QSA}$	Backend
1	114	222	114	168	114	132	114	132	AerSimulator
2	120	186	120	144	120	126	120	126	AerSimulator
3	126	192	126	192	102	192	102	162	AerSimulator
4	156	216	156	180	156	180	114	168	AerSimulator
5	168	174	168	156	168	174	162	150	AerSimulator
6	78	96	78	72	78	72	78	72	AerSimulator
7	210	288	210	288	198	282	180	276	AerSimulator
8	240	312	222	282	210	258	180	234	AerSimulator
9	234	330	234	270	210	204	192	198	AerSimulator
10	180	288	150	282	144	282	144	222	AerSimulator
11	210	276	162	234	156	174	156	174	AerSimulator
12	156	228	156	228	156	228	150	228	AerSimulator
13	246	inf	234	300	228	282	216	264	AerSimulator
14	288	444	264	348	240	306	240	306	AerSimulator
15	222	408	222	366	222	330	216	312	AerSimulator
16	246	420	234	372	234	360	234	342	AerSimulator
17	300	402	282	336	234	312	228	294	AerSimulator
18	342	438	330	360	276	318	276	318	AerSimulator
19	660	774	660	654	534	612	540*	594	Ibm Quantum Platform
20	618	702	570	690	474	576	456	588*	Ibm Quantum Platform
21	612	684	600	612	522	570	462	510	Ibm Quantum Platform
22	666	768	642	690	468	564	474*	552	Ibm Quantum Platform
23	576	708	576	690	546	618	540	600	Ibm Quantum Platform
24	606	726	594	630	510	552	474	516	Ibm Quantum Platform
36	576	618	576	606	450	480	420	456	Ibm Quantum Platform
39	666	816	666	678	534	552	534	546	Ibm Quantum Platform
42	870	966	858	750	708	636	636	666*	Ibm Quantum Platform

In the 27 instances presented in the table above, as observed, the HA-QSA and Sweep-QSA algorithms each provided weaker solutions compared to other algorithms in only two instances. In all other cases, they consistently produced solutions that were equal to or superior to those of the other algorithms. Considering the significantly shorter runtime of these quantum algorithms on quantum computers, this suggests that, with the advancement of quantum computing, these algorithms could effectively replace classical optimization algorithms, yielding higher-quality solutions for complex combinatorial optimization problems. In the images below, the quantum circuits corresponding to the QSA algorithm for Instance 1 are presented.

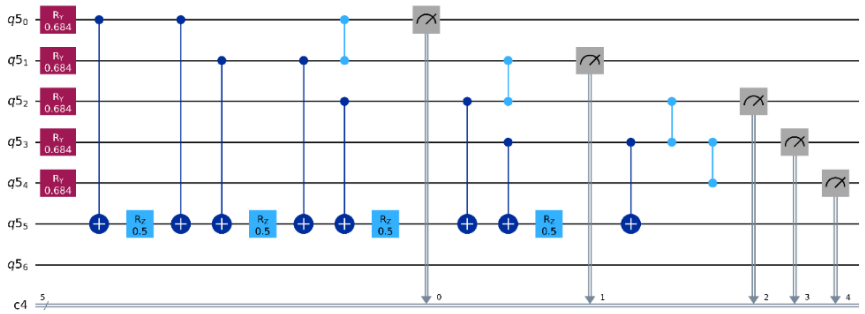
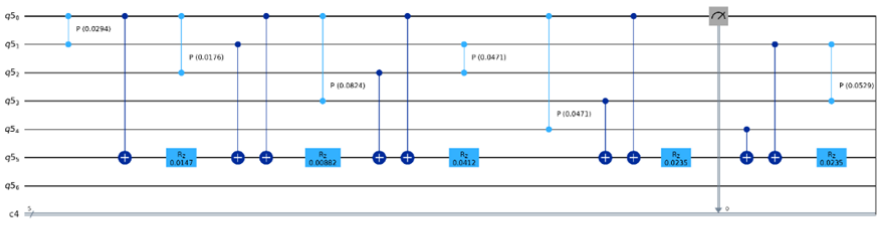


Figure 1. Mixer Circuit of Instance 1



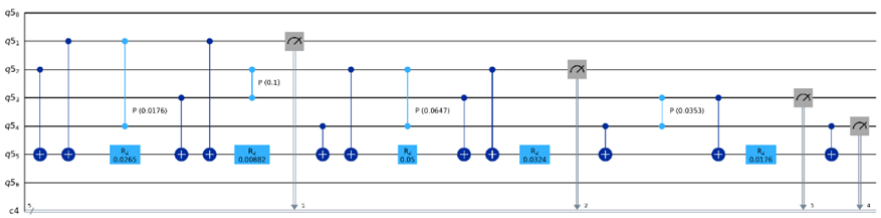


Figure 2. Phase Separator Circuit of Instance 1

The images below present a comparison of search-based algorithms, classified based on the initialization algorithms employed to obtain their initial solutions. The results indicate the superiority of quantum-based search algorithms.

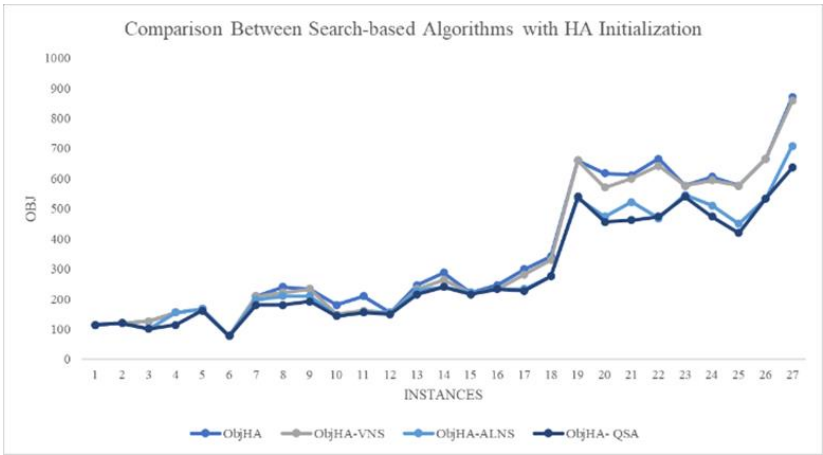
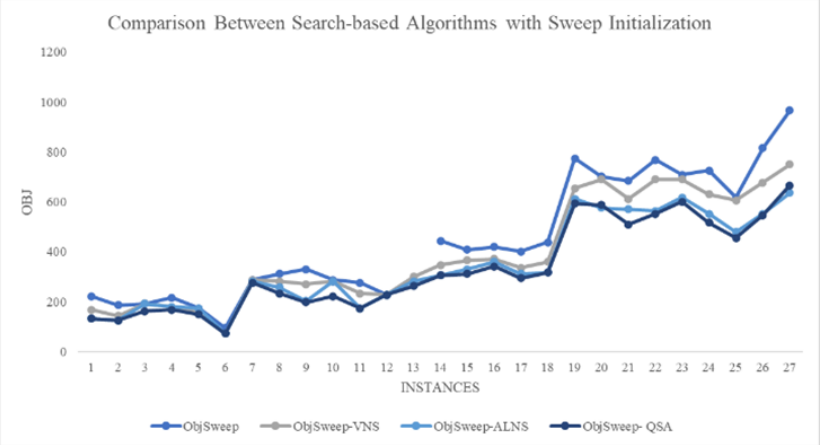


Figure 3 .Comparison Between Search-based Algorithms with HA Initialization



6. Conclusion

This study highlights the potential of hybrid quantum-classical algorithms, particularly Quantum-Enhanced Simulated Annealer (QSA), in solving complex last-mile delivery problems. By combining quantum exploration with classical optimization techniques, QSA significantly outperforms traditional local search algorithms like Variable Neighborhood Search and Adaptive Large Neighborhood Search in both solution quality and computational efficiency. Experiments on the Qiskit quantum simulator and the IBM Quantum Platform demonstrate that QSA achieves near-optimal solutions with fewer iterations, making it well-suited for large-scale combinatorial optimization. The research contributes to the growing field of quantum-enabled optimization in logistics and supply chain management. It addresses key challenges such as scalability and adaptability, showing that even with current hardware limitations, hybrid quantum-classical approaches can transform the logistics industry by reducing costs, improving service levels, and enhancing sustainability. Future research should focus on scaling these algorithms to more complex problem instances. Additionally, integrating quantum algorithms with real-time data streams will be essential for practical applications.

References

[1] F. Arute *et al.*, 'Quantum supremacy using a programmable superconducting processor', *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[2] M. Streif, S. Yarkoni, A. Skolik, F. Neukart, and M. Leib, 'Beating classical heuristics for the binary paint shop problem with the quantum approximate optimization algorithm', *Phys Rev A (Coll Park)*, vol. 104, no. 1, p. 012403, 2021.

[3] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, 'Quantum annealing for industry applications: Introduction and review', *Reports on Progress in Physics*, vol. 85, no. 10, p. 104001, 2022.

[4] R. Harikrishnakumar, S. Nannapaneni, N. H. Nguyen, J. E. Steck, and E. C. Behrman, 'A quantum annealing approach for dynamic multi-depot capacitated vehicle routing problem', *arXiv preprint arXiv:2005.12478*, 2020.
[5] M. Borowski *et al.*, 'New hybrid quantum annealing algorithms for solving vehicle routing problem', in *International Conference on Computational Science*, Springer, 2020, pp. 546–561.
[6] E. Osaba, E. Villar-Rodriguez, and A. Asla, 'Solving a real-world package delivery routing problem using quantum annealers', *Sci Rep*, vol. 14, no. 1, p. 24791, 2024.
[7] S. Feld *et al.*, 'A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer', *Frontiers in ICT*, vol. 6, p. 13, 2019.
[8] S. J. Weinberg, F. Sanches, T. Ide, K. Kamiya, and R. Correll, 'Supply chain logistics with quantum and classical annealing algorithms', *Sci Rep*, vol. 13, no. 1, p. 4770, 2023.
[9] K. Srinivasan, S. Satyajit, B. K. Behera, and P. K. Panigrahi, 'Efficient quantum algorithm for solving travelling salesman problem: An IBM quantum experience', *arXiv preprint arXiv:1805.10928*, 2018.
[10] A. Abbas *et al.*, 'Challenges and opportunities in quantum optimization', *Nature Reviews Physics*, pp. 1–18, 2024.
[11] T. Peng, A. W. Harrow, M. Ozols, and X. Wu, 'Simulating large quantum circuits on a small quantum computer', *Phys Rev Lett*, vol. 125, no. 15, p. 150504, 2020.
[12] A. Lowe *et al.*, 'Fast quantum circuit cutting with randomized measurements', *Quantum*, vol. 7, p. 934, 2023.
[13] S. Sinno *et al.*, 'Performance of Commercial Quantum Annealing Solvers for the Capacitated Vehicle Routing Problem', *arXiv preprint arXiv:2309.05564*, 2023.
[14] L. S. Herzog *et al.*, 'Improving Quantum and Classical Decomposition Methods for Vehicle Routing', *arXiv preprint arXiv:2404.05551*, 2024.
[15] F. Sanches, S. Weinberg, T. Ide, and K. Kamiya, 'Short quantum circuits in reinforcement learning policies for the vehicle routing problem', *Phys Rev A (Coll Park)*, vol. 105, no. 6, p. 062403, 2022.
[16] V. V Dixit and C. Niu, 'Quantum computing for transport network design problems', *Sci Rep*, vol. 13, no. 1, p. 12267, 2023.
[17] F. Neukart, G. Compostella, C. Seidel, D. Von Dollen, S. Yarkoni, and B. Parney, 'Traffic flow optimization using a quantum annealer', *Frontiers in ICT*, vol. 4, p. 29, 2017.
[18] B. E. Gillett and L. R. Miller, 'A heuristic algorithm for the vehicle-dispatch problem', *Oper Res*, vol. 22, no. 2, pp. 340–349, 1974.
[19] D. Reyes, M. Savelsbergh, and A. Toriello, 'Vehicle routing with roaming delivery locations', *Transp Res Part C Emerg Technol*, vol. 80, pp. 71–91, 2017.