ISSN 2464-4617 (print)
ISSN 2464-4625 (online)
Quantum Informatics, Computing & Technology 2025
QC-Horizon 2025
https://www.qc-horizon.eu/

Model-driven development of heterogeneous quantum models for multi-vehicle route planning

Furkan Polat
Koç University
fpolat17@ku.edu.tr

Hasan Tunçer
Purdue University
htuncer@purdue.edu

Hamed Gholipour
University of Beira Interior
hamed.gholipour@ubi.pt

Joao Neves
University of Beira Interior
jcneves@ubi.pt

Moharram Challenger
University of Antwerp & Flanders Make
moharram.challenger@uantwerpen.be

ABSTRACT

Quantum computing has the potential to significantly improve large-scale optimization, particularly in logistics. However, practical adoption remains limited due to fragmented hardware ecosystems and the complexity of quantum programming. We present **MDE4QC**, a model-driven engineering framework that introduces a Platform-Independent Model (PIM) for defining routing problems at a high level of abstraction. The framework automatically transforms these models into executable code for quantum annealers, gate-based quantum systems, and classical solvers. MDE4QC is integrated with cloud platforms such as D-Wave and IBM Qiskit and supports hybrid execution flows. An intuitive graphical interface enables users to define routing problems without writing quantum or classical code. Users simply select the target platform and configure problem parameters through a user-friendly interface; the framework handles model generation, transformation, and execution. We validate MDE4QC using open-access real logistic data from the City of Antwerp to demonstrate its ability to reduce development effort, ensure cross-platform portability, and deliver measurable gains in routing efficiency.

Keywords

Quantum computing, model-driven engineering, platform-independent modeling, route optimization, quantum annealing, hybrid solvers, domain-specific modeling language, logistics

1 INTRODUCTION

Quantum computing holds great promise for solving large-scale optimization problems [1, 2], especially in fields like logistics and transportation [3]. However, its adoption in real-world applications remains limited [4]. One of the key challenges is the heterogeneity of the current quantum ecosystem: Different hardware providers, programming interfaces, execution pipelines, and cloud platforms make development complex and fragmented.

Model-Driven Engineering (MDE) has a proven history in classical software engineering to reduce complexity by enabling users to work at a higher level of abstraction [5, 6, 7]. It simplifies development through automatic code generation and platform-independent design.

In this paper, we use MDE4QC, a framework that brings the principles of MDE to quantum computing [8, 9]. MDE4QC allows users to define routing problems using a hardware-agnostic model, which can then be automatically transformed into executable code compatible with various quantum and hybrid backends. Through a case study on multi-vehicle route planning, we demonstrate how this approach can make quantum-enhanced optimization more accessible to logistics practitioners without requiring deep expertise in quantum computing.

1.1 Motivation and Problem Statement

Modern logistics increasingly relies on solving high-frequency, large-scale routing problems with high precision. This is pivotal not only for reducing operational costs and ensuring timely deliveries but also for addressing larger systemic issues such as fuel efficiency and urban traffic congestion. As the number of vehicles, operational constraints, and dynamic environmental factors grow, the complexity of these optimization problems often exceeds the capabilities of traditional computational methods [10].

Quantum computing presents a promising frontier for addressing such combinatorial optimization problems. Yet, the current quantum ecosystem is highly fragmented. The coexistence of multiple hardware vendors, inconsistent programming interfaces, and heterogeneous execution models introduces substantial technical overhead [11]. For logistics professionals lacking a background in quantum technologies, these inconsistencies pose a significant barrier to effective adoption. Quantum cloud services have made it easier to access quantum hardware, but variations in how each platform works and the different programming languages they use still make it difficult to combine them effectively. Implementing and maintaining quantum algorithms across various providers still demands advanced expertise and often leads to redundant development efforts.

This context underscores the value of a model-driven approach. Our proposed framework, **MDE4QC** (Model-Driven Engineering for Quantum Computing), mitigates these challenges by providing platform-independent abstractions, automated model-to-model transformations, and infrastructure-agnostic execution pathways. This enables logistics experts to specify routing problems using high-level modeling constructs, while the system transparently handles the synthesis and execution of suitable quantum or hybrid solutions based on the current computational resources.

Crucially, the benefits of MDE4QC extend beyond computational performance. By enabling more efficient routing, the framework contributes to the reduction of fuel consumption and greenhouse gas emissions—particularly vital in urban contexts where smart routing directly affects environmental sustainability. Moreover, by integrating live data streams such as traffic conditions and weather forecasts, MDE4QC could facilitates adaptive, eco-aware decision-making. These capabilities support broader policy objectives, such as those outlined in the European Green Deal, and emphasize the framework's societal impact and alignment with sustainable development goals [12].

1.2 Objectives and Contributions

This paper makes the following key contributions to the field of quantum-enabled logistics optimization:

1. MDE4QC Framework: Transforms high-level logistics problems into executable solutions for classical, quantum, and hybrid platforms. Model Transformation of this framework converts abstract models into validated, optimized, and deployable code across platforms [13].

2. Platform-Agnostic Modeling: Uses a domain-specific language to support classical (e.g., K-Means, GNN), annealing, and gate-based quantum methods [14]. This modelling language allows users to define routing problems graphically, with no coding required (no code approach) [15].

3. Solver Selection: Chooses the best solver based on problem size, hardware availability, and cost-performance trade-offs. The cloud integration of the framework connects with major quantum and classical platforms as solvers for automated execution and result collection [16].

4. The City of Antwerp Case Study: Validated on real routing data from the City of Antwerp to benchmark solver performance [17]. It uses live traffic and environmental data to cut fuel use and emissions [18].

This paper is organised as follows: Section 2 discusses the related work in the literature. The fundamentals and background for this study are elaborated in Section 3. Section 4 presents the architecture for the proposed framework and discusses its implementation details. The framework is evaluated in section 5 using a case study, and the results are discussed in section 6. Finally, the paper is concluded in section 7.

2 RELATED WORK

In recent years, the convergence of quantum computing and route optimization has garnered substantial interest, driven by the need to address scalability challenges that classical methods often encounter. Traditional techniques for solving the Multi-Vehicle Routing Problem (MVRP), including mixed-integer programming, evolutionary strategies, and more recently, machine learning-based approaches such as Graph Neural Networks (GNNs), have achieved notable success [19, 20]. Nevertheless, these models frequently face difficulties in handling high-dimensional, constraint-rich scenarios, especially in real-time urban logistics environments.

Quantum computing introduces a new paradigm for tackling combinatorial problems by Utilizing unique properties such as superposition and entanglement. Notable frameworks, including the Quantum Approximate Optimization Algorithm (QAOA) [21] and the Binary Quadratic Model (BQM) utilized in D-Wave systems [22], have shown promising results in constrained optimization tasks. Fitzek et al. [23] demonstrated the effectiveness of QAOA in heterogeneous vehicle routing scenarios, underscoring the advantages of quantum speedup under certain conditions. Further studies by Willsch et al. [24] and Neukart et al. [25] applied quantum annealing to traffic-aware routing, although their implementations were limited by current hardware constraints and embedding overheads. A central barrier to the broader adoption of quantum approaches lies in the fragmented nature of the quantum computing landscape. Each platform typically requires distinct programming interfaces and problem formulations, which complicates the development process and inhibits portability. To mitigate this, researchers have increasingly adopted Model-Driven Engineering (MDE) techniques to introduce abstraction layers and reduce platform dependency. One early example is the MDE4QP framework [26], which applied model-driven principles to quantum chemistry, facilitating code generation for both gate-based and annealing architectures through automated transformations.

This paper builds upon that foundation by presenting MDE4QC, a model-driven framework specifically designed for quantum-enhanced logistics optimization. MDE4QC introduces a domain-specific modeling language (DSML) and transformation pipeline that supports both classical and quantum solvers, including IBM Qiskit and D-Wave. Unlike prior tools, MDE4QC allows users to define problems through a visual interface and execute solutions across heterogeneous platforms without requiring low-level quantum programming expertise. Empirical validation using open urban data further emphasizes the framework's applicability to real-world logistics problems.

MDE4QC supports cloud-native deployment via an interactive dashboard and RESTful APIs. The API is designed to let external systems bypass the UI and communicate directly with solvers, enabling automation and integration in industrial workflows. The framework aligns with recent advances in hybrid quantum-classical computing [27] and supports scalable, interoperable solutions for smart cities and complex logistics systems.

3 FUNDAMENTALS

This section elaborates on the fundamental topics required to understand this paper, including Model-Driven Engineering (MDE), MDE for Quantum Computing, Platform Independent Modelling, QAOA algorithm and quantum anealling samplers.

3.1 Model-Driven Engineering (MDE)

Model-Driven Engineering (MDE) is a software development paradigm that prioritizes the use of high-level, abstract models as the core artifacts throughout the engineering process. Instead of relying primarily on manually written source code, MDE emphasizes the use of domain-specific modeling languages (DSMLs) to represent the key aspects of a system or problem domain. These models become the basis for both system design and implementation [28, 29].

By elevating the level of abstraction, MDE facilitates clearer communication among stakeholders, supports early validation of system designs, and promotes the reuse of established architectural patterns. Through automated model-to-model and model-to-code transformations, MDE streamlines code generation and minimizes manual implementation errors. This methodology has demonstrated particular value in complex and heterogeneous environments, including embedded systems, enterprise platforms, and automotive domains such as AUTOSAR [30, 31].

3.2 MDE for Quantum Computing (MDE4QC)

Quantum computing platforms are inherently diverse. Gate-based systems and quantum annealers represent two fundamentally different approaches to quantum computation. In addition, real-world systems are deployed across hardware with distinct capabilities, connectivity constraints, and programming interfaces. This variability creates a fragmented ecosystem, making development more difficult and reducing the portability of algorithms.

Our previous work on ground state energy calculations introduced a unified framework, MDE4QP, that uses model-driven abstractions to bridge the gap between quantum annealing and gate-based models. That framework demonstrated how domain-specific abstractions could be translated into platform-specific implementations via systematic model-to-model and model-to-code transformations, allowing the same solution to be expressed and executed across both types of quantum device [8].

In this paper, we extend that vision to a new application domain, multivehicle routing optimization, where quantum computing shows promise but suffers from the same fragmentation and entry barriers. MDE4QC, our updated framework, enables users to define routing problems at a high level using a platform-independent modeling language. These models are then automatically transformed into quantum formulations suited for specific architectures, such as Binary Quadratic Models (BQM) for D-Wave annealers or Qubit-Hamiltonian expressions for variational quantum algorithms (VQAs) on platforms like Qiskit or Cirq.

This abstraction allows algorithm designers and logistics practitioners to express optimization problems without needing deep knowledge of quantum physics, hardware, or SDKs. Moreover, it supports cross-platform evaluation, facilitating empirical comparisons between quantum annealing and gate-based solutions on real-world routing problems [9].

Building on insights from our prior research, this work prioritizes hardware-agnostic modeling, automated transformation pipelines, and a standardized development lifecycle. The result is a robust framework that not only simplifies quantum software engineering but also brings practical quantum optimization closer to adoption in industrial and urban logistics.

3.3 Quantum Approximate Optimization Algorithm (QAOA)

The Quantum Approximate Optimization Algorithm (QAOA) is a hybrid quantum-classical approach designed to solve combinatorial optimization problems by alternating quantum operators derived from problem-specific cost and mixing Hamiltonians [21]. In this work, we integrate QAOA with Graph Neural Networks (GNNs) to address traffic route optimization in urban transportation networks. The GNN models dynamic traffic conditions using spatiotemporal data and updates edge weights representing congestion and travel times [32]. These predicted weights are encoded into a cost Hamiltonian that guides the QAOA circuit

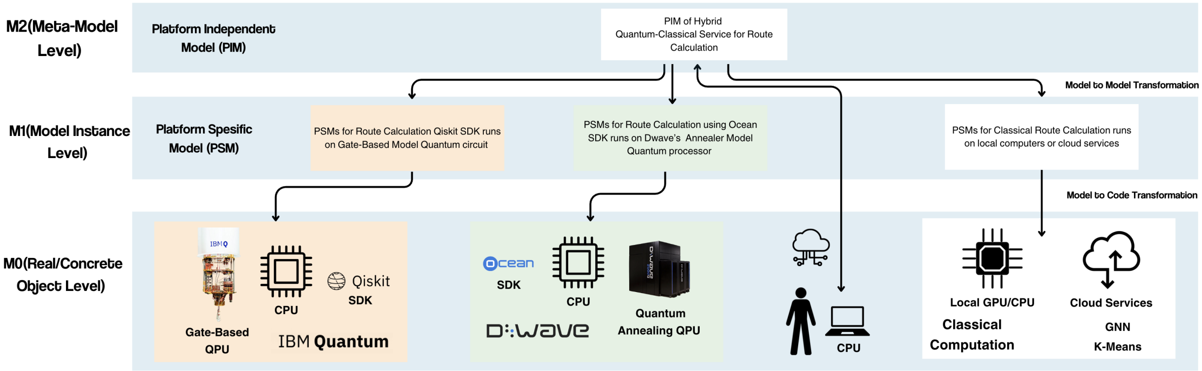

Figure 1: MDE4QC Framework for Hybrid Quantum-Classical Route Optimization. The PIM layer defines platform-independent route optimization models, while the PSM layer specifies platform-specific implementations using Qiskit (IBM Quantum), Ocean SDK (D-Wave), or classical methods (GNN, K-Means).

toward optimal routing decisions [33]. The hybrid framework leverages GNNs for learning accurate traffic patterns and QAOA for efficient exploration of optimal paths via quantum state evolution. This integration offers a scalable and adaptive solution for real-time route planning, logistics optimization, and intelligent transportation systems, particularly suited for deployment on near-term quantum devices.

3.4 Quantum Annealing Samplers

Quantum annealing samplers serve as critical interface engines that facilitate practical implementation of adiabatic quantum computation for large-scale optimization problems [34]. The `LeapHybridNLSampler` specializes in non-linear optimization formulations, automatically performing problem decomposition and variable embedding to handle complex routing constraints [35]. The `LeapHybridDQMSampler` operates on Discrete Quadratic Model formulations, mapping discrete decision variables to quantum annealing problems with penalty-based constraint enforcement [36]. These samplers implement intelligent partitioning strategies that identify subproblems suitable for quantum acceleration while delegating preprocessing tasks to classical algorithms [37]. The hybrid architecture automatically handles quantum embedding, error mitigation, and solution validation, making quantum annealing accessible for enterprise-scale logistics optimization [38].

3.5 Platform Independent Modelling

Cloud-based quantum computing platforms such as IBM Qiskit, Google Cirq, Amazon Braket, and D-Wave Leap enable users to execute quantum algorithms remotely via their respective SDKs [39, 40, 41, 42]. These environments primarily support Python, but are increasingly extending compatibility to other programming languages including C++, JavaScript, and Q# through RESTful APIs, Jupyter notebook interfaces, and language bindings. Additionally, they offer graphical user interfaces, quantum simulators, and visualization tools that facilitate both beginner-level experimentation and the development of advanced quantum applications.

Despite these advancements, a critical challenge to the practical adoption of quantum computing remains: platform fragmentation. Each provider defines distinct programming paradigms, toolchains, and circuit specification interfaces. As a result, developers must master multiple APIs and re-implement problem logic to accommodate each backend, posing a significant barrier to entry and scalability [43, 44].

To address this limitation, we propose the **Model-Driven Engineering for Quantum Computing (MDE4QC)** framework, which introduces a **Platform-Independent Modeling (PIM)** methodology specifically adapted for quantum applications. Rather than writing platform-specific code, users define high-level, platform-agnostic models of their problems—such as a multi-vehicle routing scenario. These abstract specifications are then automatically transformed into **Platform-Specific Models (PSMs)** tailored to the syntax and semantics of the target quantum backends. For instance, a single PIM model can be mapped to a variational quantum circuit in Qiskit or translated into a quantum annealing formulation compatible with D-Wave's architecture. The overall architecture of the MDE4QC framework is illustrated in Figure 1.

This model-driven approach abstracts away the need for backend-specific programming skills. Domain parameters—such as the number of vehicles, route constraints, cost objectives, or Ising model representations—are captured declaratively within the PIM. They are subsequently compiled into executable code via automated model transformations. Tasks that traditionally required extensive manual coding and SDK-specific integration are now streamlined into a fully automated workflow.

In our prior work, we demonstrated this capability by deploying a unified PIM for ground state energy estimation on both Qiskit and D-Wave platforms without modifying the original model [10]. Here, we extend this methodology to **quantum-enhanced logistics optimization**, enabling seamless execution across heterogeneous quantum environments [45].

4 ARCHITECTURE AND IMPLEMENTATION

The system uses a modular, layered architecture that clearly separates different tasks and allows easy integration of classical and quantum optimization methods. Each layer can be extended independently and communicates with others through defined interfaces (see Figure 2 for an overview).

The development of this framework involved adapting key back-end-specific implementations from existing open-source projects. In particular, the routing optimization component for the D-Wave back-end was originally derived from the *D-Wave Ocean SDK example repository*, which provided a robust foundation for quantum annealing-based formulations [46]. We gratefully acknowledge D-Wave Systems Inc. for making these examples publicly available. Furthermore, our complete implementation, including model transformations and platform-specific templates, is made openly accessible via the *MDE4QC GitHub repository* [47].

4.1 Solvers Group

The solvers group represents the computational center of the system, divided into quantum and classical components. The quantum solvers utilize both D-Wave's annealing technology and IBM's gate-based quantum computing capabilities. The D-Wave integration includes specialized samplers for different problem types, while the IBM Quantum integration provides QAOA implementation with configurable optimization parameters.

The D-Wave implementation employs two hybrid quantum-classical samplers: `LeapHybridNLSampler` and `LeapHybridDQMSampler`. The former targets non-linear optimization problems such as complex routing scenarios, while the latter is suited for Discrete Quadratic Models (DQMs). Both samplers automate problem decomposition, embedding, and qubit mapping, enabling scalable optimization with built-in solution validation.

The IBM Quantum pipeline applies QAOA via Qiskit to solve the Capacitated Vehicle Routing Problem (CVRP). The problem is formulated using binary variables within the `QuadraticProgram` class, converted to a QUBO, and optimized using the `MinimumEigenOptimizer` with COBYLA (100 iterations, `reps=2`). Final routes are extracted using NetworkX.

Classical solvers complement the quantum components by offering GNN-based route prediction and K-Means clustering. The GNN models learn from historical patterns to propose effective routing schemes [48], while K-Means efficiently partitions the location space [49]. A dynamic solver selection mechanism chooses the most appropriate method based on problem characteristics, ensuring robust performance across different scales.

4.2 Data Layer

The data layer bridges the gap between raw logistics data and optimization engines, with support tailored for the City of Antwerp Open Data Portal [50]. Datasets include structured location data such as public bike parking, facilities, and service points in CSV format. The system imports these datasets directly, extracting object IDs, coordinates, and names.

To ensure compatibility, the system performs coordinate transformations from EPSG:3857 to EPSG:4326 and validates all required fields. This preprocessing phase ensures that incoming data is fully compatible with the optimization solvers. Technologies used include `osmnx` for road network graph generation, `networkx` for graph operations, and `pyproj` for coordinate transformations. `Pandas` handles tabular data management, while `folium` enables map-based visualization. Diskcache is employed for local result caching to enhance responsiveness.

The system also features a user-friendly Dash interface. Users can upload files through a drag-and-drop interface or select from existing entries in the `assets/CSVs` directory. Uploaded files are automatically processed, including coordinate transformation and route metadata extraction, readying the data for downstream optimization.

4.3 External Dependencies

The system relies on several external packages and SDKs. Web-based visualization and interaction are managed through `Dash`, `Folium`, and `OSMnx`. Quantum capabilities are supported via the D-Wave Ocean SDK (including `dimod` for problem formulation) and the IBM Qiskit stack (including `qiskit-algorithms` and `qiskit-optimization`). Classical machine learning functionality is implemented using `PyTorch` and `PyG`, while logistics graph processing and coordinate transformations are handled with `NetworkX`, `PyProj`, `Pandas`, `NumPy`, and `SciPy`. Performance optimization utilizes `diskcache` for result caching and `multiprocess` for parallel execution. The system integrates with ERP and WMS platforms for

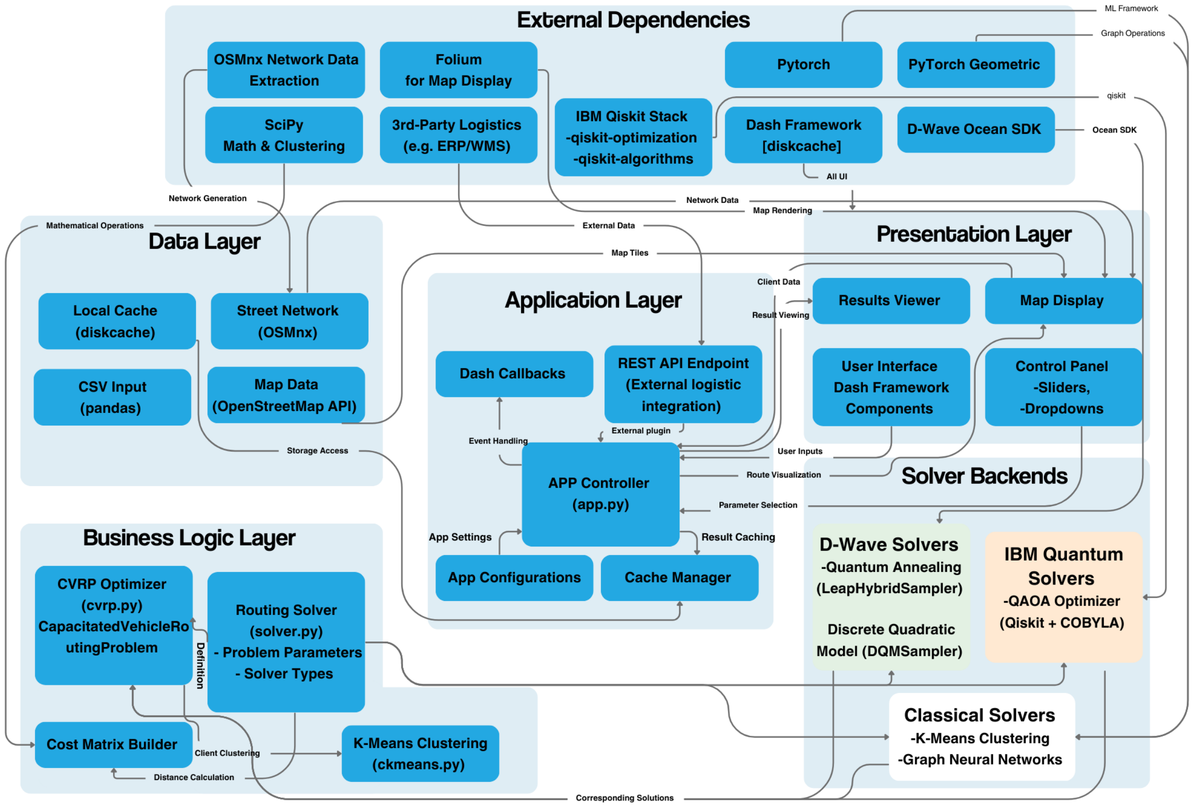

Figure 2: System architecture of the MDE4QC framework for hybrid quantum-classical route optimization. The architecture is organized into layered modules: the Data Layer handles geographic inputs and caching, the Application Layer manages user interaction and API calls, and the Business Logic Layer defines solver logic. Solver Backends support quantum (D-Wave and IBM Qiskit) and classical (K-Means, GNN) algorithms. External Dependencies provide libraries and SDKs used across all layers. Inter-layer communication flows are shown, including event handling, parameter passing, and result rendering.

real-time logistics data and utilizes the OpenStreetMap API for additional geographic information.

5 CASE STUDY: MULTI-VEHICLE ROUTE PLANNING

This section presents the real-world application of our MDE4QC framework to the Multi-Vehicle Route Planning (MVRP) problem, implemented and tested using real map data from the City of Antwerp [1]. The problem builds upon the Capacitated Vehicle Routing Problem (CVRP), a classical NP-hard optimization task that requires assigning client visits to a fleet of vehicles such that each vehicle respects its capacity constraint and the overall travel distance is minimized [51].

Unlike simplified formulations, our implementation captures the spatial and logistical complexity of real-world transport systems [52]. It includes detailed geographic representations, multiple vehicle types, depot configurations, and hybrid cost models based on either road network routing or Euclidean geometry.

Figure 3 illustrates the types of scenarios designed to support by our system. These range from conventional single-depot delivery routing to advanced use cases like dynamic route recalculation in real-time or handling heterogeneous fleets. These variations require different encoding strategies, cost models, and solver configurations at both the Platform-Independent Model (PIM) and Platform-Specific Model (PSM) levels.

The City of Antwerp dataset consists of real geo-coordinates, mapped to the nearest road nodes using OSMnx [52]. Depending on vehicle type, vehicles are routed across this network using Dijkstra's algorithm or Euclidean metrics. Capacity constraints are handled either via penalty encoding in quantum solvers or direct enforcement in classical logic [53].

A brute-force approach is infeasible due to factorial scaling in clients and exponential scaling in vehicles. To overcome this, our modular solver architecture supports classical heuristics (e.g., K-Means [54]) and

[1] https://portaal-stadantwerpen.opendata.arcgis.com/

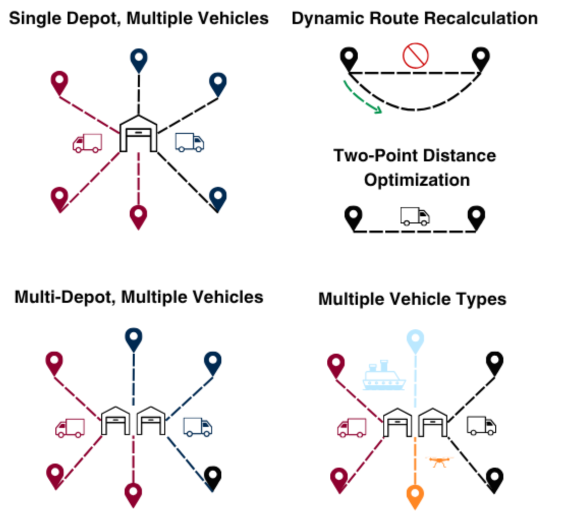

Figure 3: Optimization scenarios modeled within the MDE4QC framework. The figure highlights key variations in routing complexity: depot configurations (single vs. multiple), vehicle heterogeneity (trucks, drones, boats), and dynamic considerations such as route recalculation and short-distance delivery optimization.

quantum algorithms like QAOA [21] and LeapHybrid solvers [55].

6 RESULTS AND DISCUSSION

This section presents empirical evaluation of the MDE4QC framework through benchmarks comparing quantum and classical solvers. Our goal was to measure solver efficiency and provide information for strategic decision-making.

Evaluation metrics:

- Total distance (m): Sum of all routes.
- Computation time (s): Optimization duration.
- Load distribution: Client-to-vehicle balance.
- Scalability: Solver response to increased client locations.

Benchmarking approach

We tested D-Wave hybrid quantum annealing and classical K-Means clustering across 18 scenarios (10–2013 clients). For each, we recorded:

- Objective distance
- Execution time

Regression models derived from these results guide future solver selection [56].

Distance and Time Projections

We applied cubic polynomial regression:

$$D(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

to fit total distance and time metrics. Figures 4 and 5 depict the results.

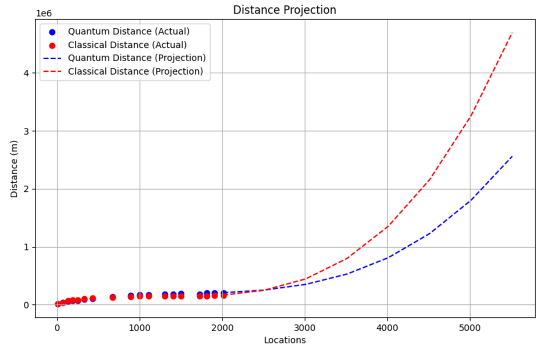

Figure 4: Total distance vs. client locations. Polynomial regression forecasts quantum (Blue line, bottom) vs. classical (Red line, top) crossover points.

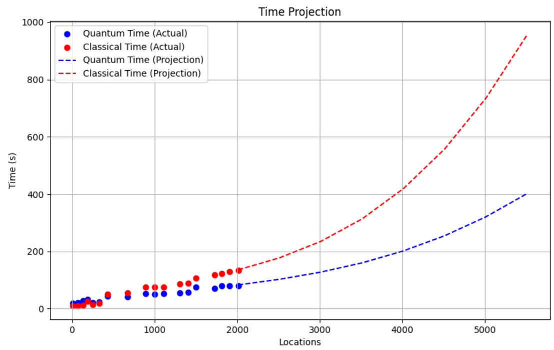

Figure 5: Computation time vs. client locations. Classical solvers scale better beyond 1300 clients.

Solver Decision Model

We developed a cost-based decision model to determine when quantum solvers are preferable over classical ones. The model compares distance savings against computational time costs to guide solver selection in a practical, scenario-dependent way.

Quantum solvers generally produce shorter total route distances, which reduces fuel consumption and operational costs. However, they are slower and more expensive to run. Classical solvers are faster and cheaper but often generate longer routes. The model calculates the net benefit using:

$$\Delta Cost = (D_C - D_Q) \cdot c_d - (T_Q - T_C) \cdot c_t$$

where:

- D_C, D_Q: Route distances from classical and quantum solvers,
- T_C, T_Q: Solver runtimes,
- c_d: Cost per kilometer,
- c_t: Cost per unit time.

A positive $\Delta Cost$ means the quantum solver offers financial advantage.

We implemented this model in a spreadsheet to allow users to adjust inputs such as delivery count, distance cost, and compute cost. Solver behavior is estimated using regression equations derived from empirical test runs. This makes the model usable without requiring programming or domain expertise.

Discussion

The implementation of the MDE4QC framework underscores the evolving potential of hybrid quantum-classical optimization in real-world logistics. While the empirical advantages of hybrid solvers—in the scope of this study D-Wave's LeapHybrid models—are evident in medium-scale routing scenarios, their practical deployment raises important considerations. Notably, quantum solvers demonstrate superior performance in exploring complex solution spaces, yet this comes at the cost of increased runtime and hardware resource consumption, which may not be justifiable in time-sensitive or cost-sensitive contexts [57].

The inclusion of a decision-support mechanism within MDE4QC represents a step forward in balancing these trade-offs. By quantifying key parameters—such as route efficiency, execution delay, and computational cost—it shifts solver selection from heuristic choice to evidence-based reasoning.

Despite the framework's support for platform-independent modeling, full integration with gate-based quantum platforms remains incomplete. This reflects a challenge in the NISQ era, where hardware limitations constrain the real-world applicability of algorithms like variational quantum algorithms (VQAs) [27]. While promising, the utility of gate-based methods in logistics contexts remains largely theoretical and awaits further maturity in both software and hardware ecosystems.

An important aspect of this study is the framework's contribution to sustainability. MDE4QC helps reduce travel distance and fuel usage, which supports environmentally friendly logistics. This aligns with goals such as the European Green Deal. As a result, the framework adds value not only through technical improvements but also by promoting greener operations.

Ultimately, while MDE4QC demonstrates how model-driven engineering can mitigate the fragmentation of the quantum computing ecosystem, its broader adoption will depend on improvements in backend compatibility, user interface integration, and adaptive capabilities in dynamic logistics settings.

7 CONCLUSION AND FUTURE WORK

The MDE4QC framework delivers a robust model-driven solution for quantum-enhanced logistics optimization by abstracting complex routing problems into high-level, platform-independent models. These are automatically transformed into executable formats suitable for both quantum and classical backends. With its hybrid solver architecture and intelligent decision-making mechanism, MDE4QC enables efficient and adaptable route planning. Validation using real-world data from the City of Antwerp confirms its effectiveness in improving routing performance and supporting environmental sustainability. While current limitations of gate-based quantum hardware remain, the framework marks a significant step toward scalable and interoperable quantum solutions for real-world logistics. Future enhancements will target broader platform integration and real-time dynamic responsiveness.

8 REFERENCES

[1] R. Orús, S. Mugel, and E. Lizaso, "Quantum computing for finance: Overview and prospects," *Reviews in Physics*, vol. 4, p. 100028, 2019.

[2] A. Ajagekar and F. You, "Quantum computing for energy systems optimization: Challenges and opportunities," *Energy*, vol. 179, pp. 76–89, 2019.

[3] E. Osaba, E. Villar-Rodriguez, and I. Oregi, "A systematic literature review of quantum computing for routing problems," *IEEE Access*, vol. 10, pp. 55805–55817, 2022.

[4] E. National Academies of Sciences and Medicine, *Quantum Computing: Progress and Prospects*. Washington, DC: The National Academies Press, 2019.

[5] D. Schmidt, "Guest editor's introduction: Model-driven engineering," *Computer*, vol. 39, no. 2, pp. 25–31, 2006.

[6] M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in Practice*, vol. 1. 09 2012.

[7] R. France and B. Rumpe, "Model-driven development of complex software: A research roadmap," in *Future of Software Engineering (FOSE '07)*, pp. 37–54, 2007.

[8] F. Polat, H. Tuncer, A. Moin, and M. Challenger, " Model-Driven Engineering for Quantum Programming: A Case Study on Ground State Energy Calculation ," in *2024 IEEE 48th Annual*

Computers, Software, and Applications Conference (COMPSAC), pp. 2353–2360, 2024.

[9] A. Moin, M. Challenger, A. Badii, and S. Günnemann, "Towards model-driven engineering for quantum ai." INFORMATIK 2022, 2022.

[10] A. Moin, "Quantum-enhanced logistics: A model-driven framework for urban optimization," *Journal of Advanced Computational Logistics*, 2023.

[11] T. Chalenger, "The fragmented landscape of quantum computing platforms: Implications for industrial applications," *Quantum Systems and Engineering*, vol. 9, no. 2, pp. 117–132, 2024.

[12] L. Wibert, "Sustainable logistics through adaptive quantum systems: Aligning technology with environmental goals," *Sustainable Computing: Informatics and Systems*, 2024.

[13] H. Moin and T. El-Sayed, "Model-driven transformation pipelines for quantum-classical systems," *Software and Systems Modeling*, vol. 21, no. 5, pp. 889–911, 2022.

[14] B. Chalenger and M. Torres, "Platform-agnostic dsml for quantum-classical logistics optimization," *ResearchGate*, 2023.

[15] A. R. Nightingale, "Visual modeling interfaces in quantum-enabled supply chain management." `https://medium.com/@arwnightingale`, 2024.

[16] D.-W. S. Inc., "Hybrid solver service overview." `https://www.dwavesys.com`, 2023.

[17] R. Tanaka and M. de Vries, "Case study: Applying hybrid quantum algorithms to smart city routing in antwerp," *Quantum Zeitgeist*, 2025.

[18] L. Fernandez and P. Müller, "Sustainability-aware logistics with quantum and classical co-processing," *QuantumRise Journal*, vol. 3, no. 1, pp. 45–59, 2025.

[19] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," *arXiv preprint arXiv:1611.09940*, 2016.

[20] E. B. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song, "Learning combinatorial optimization algorithms over graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[21] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

[22] D.-W. Systems, "Binary quadratic model (bqm) overview." `https://docs.dwavesys.com/docs/latest/doc_oceandoc_bqm.html`, 2021. Accessed: 2025-05-21.

[23] F. H. P. Fitzek *et al.*, "Applying quantum computing to the heterogeneous vehicle routing problem," *IEEE Access*, vol. 9, pp. 130146–130157, 2021.

[24] D. Willsch, M. Willsch, H. De Raedt, and K. Michielsen, "Benchmarking advantage quantum processor using qubo problems," *Quantum Information Processing*, vol. 19, no. 12, pp. 1–24, 2020.

[25] 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.

[26] M. Challenger and A. Moin, "Model-driven engineering for quantum platforms: The mde4qp approach," *Software and Systems Modeling*, vol. 22, no. 2, pp. 389–410, 2023.

[27] D. J. Egger, J. Mareček, and S. Woerner, "Hybrid quantum–classical algorithms and quantum error mitigation," *Nature Reviews Physics*, vol. 3, no. 12, pp. 734–748, 2021.

[28] D. C. Schmidt, "Model-driven engineering," *Computer*, vol. 39, no. 2, pp. 25–31, 2006.

[29] M. Brambilla, J. Cabot, and M. Wimmer, *Model-driven software engineering in practice*. Morgan Claypool, 2017.

[30] S. Sendall and W. Kozaczynski, "Model transformation: the heart and soul of model-driven software development," *IEEE software*, vol. 20, no. 5, pp. 42–45, 2003.

[31] B. Becker and et al., "Model-based development of embedded systems," in *Proceedings of the 6th ACM IEEE International conference on Embedded software*, pp. 135–144, ACM, 2006.

[32] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2020.

[33] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, no. 2, p. 34, 2019.

[34] C. C. McGeoch, "Adiabatic quantum computation and quantum annealing: Theory and practice," *Synthesis Lectures on Quantum Computing*, vol. 5, no. 2, pp. 1–93, 2014.

[35] A. Zaribafiyan, D. J. Marchand, and S. S. Changiz Rezaei, "Advantages of unfair quantum ground-state sampling," *Physical Review A*, vol. 95, no. 4, p. 042306, 2017.

[36] A. Lucas, "Ising formulations of many np problems," *Frontiers in Physics*, vol. 2, p. 5, 2014.

[37] K. Boothby, P. Bunyk, J. Raymond, and A. Roy, "Next-generation topology of d-wave quantum processors," *arXiv preprint arXiv:2003.00133*, 2020.

[38] I. Hen, J. Job, T. Albash, T. F. Rønnow, M. Troyer, and D. A. Lidar, "Probing for quantum speedup in spin-glass problems with planted solutions," *Physical Review A*, vol. 92, no. 4, p. 042325, 2015.

[39] IBM Quantum, "Qiskit: An open-source framework for quantum computing," 2023.

[40] Google AI Quantum, "Cirq: A python framework for creating, editing, and invoking Noisy Intermediate Scale Quantum (NISQ) circuits," 2023.

[41] Amazon Web Services, "Amazon Braket Documentation," 2023.

[42] D-Wave Systems Inc., "Leap: The Quantum Cloud Service," 2023.

[43] A. McCaskey and et al., "Quantum chemistry as a benchmark for near-term quantum computers," *npj Quantum Information*, vol. 5, p. 99, 2019.

[44] G. G. Guerreschi and M. Smelyanskiy, "Platform-agnostic quantum programming: Abstracting qubit connectivity and error rates," *arXiv preprint arXiv:1810.02338*, 2018.

[45] A. Challenger and M. Liu, "Visual analytics for quantum logistics optimization," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2022.

[46] h. D-Wave Ocean code examples year=2024, "D-wave ocean sdk example projects."

[47] h. MICSS-Lab year=2024, "Mde4qp github repository."

[48] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[49] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[50] C. of Antwerp, "Open data portal, city of antwerp." `https://portaal-stadantwerpen.opendata.arcgis.com/search?collection=dataset/`, 2022.

[51] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications*. SIAM, 2002.

[52] G. Boeing, "Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Computers, Environment and Urban Systems*, vol. 65, pp. 126–139, 2017.

[53] J. S. Otterbach *et al.*, "Unsupervised machine learning on a hybrid quantum computer," *arXiv preprint arXiv:1712.05771*, 2017.

[54] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[55] D.-W. Systems, "D-wave hybrid solver service whitepaper," tech. rep., 2023. `https://www.dwavesys.com/whitepapers`.

[56] T. Willems *et al.*, "Quantum optimization in real-world logistics," *Nature Quantum Applications*, 2022.

[57] T. Willems *et al.*, "Quantum optimization in real-world logistics," *Nature Quantum Applications*, 2022.