scientific reports

Check for updates

OPEN

Mini-scale traffic flow optimization: an iterative QUBOs approach converting from hybrid solver to pure quantum processing unit

Hadi Salloum[1,2,4]✉, Sanzhar Zhanalin[1], Amer Al Badr[1,3,4] & Yaroslav Kholodov[1,4]

Traffic congestion continues to pose a significant challenge in urban environments, necessitating innovative approaches to traffic management. This paper explores the application of Quantum Annealing (QA) for real-world traffic optimization, expanding on the pioneering work of Volkswagen and D-Wave. In 2017, a collaborative team demonstrated the potential of QA to optimize traffic flow by solving a complex Quadratic Unconstrained Binary Optimization (QUBO) problem involving 418 cars, which required 1,254 qubits. Later, this research culminated in a pilot project at the Web Summit conference in Lisbon, one of Europe's largest technology events, showcasing quantum computing-based traffic optimization. Since the QPU alone could not directly handle the full problem size, the team employed a hybrid classical-quantum approach, leading to significant improvements in traffic distribution. This paper builds on that foundation by investigating potential speedups using a purely quantum approach, particularly by utilizing the QPU for smaller QUBO problems. The proposed method (MTF) enhances traffic management by decomposing the overall optimization problem into smaller, more manageable subproblems. This decomposition enables us to harness the advantages of the QPU while tackling more complex traffic scenarios that previous approaches struggled to manage. By breaking the problem into smaller parts, we mitigate the challenges associated with embedding large-scale problems into the QPU, which often presents computational difficulties. To evaluate our approach, we conducted experiments involving 100, 200, 300, 400, and 500 cars on a complex traffic map featuring multiple start and end points. We successfully embedded the problem into the D-Wave Advantage Quantum Processing Unit, utilizing the "Pegasus" topology, which resulted in a significant acceleration of the solution process. The experiment results show improved speed and effectiveness in real-world scenarios by leveraging the QPU for better traffic optimization.

The recent milestones achieved by D-Wave Inc. in Quantum Annealing (QA) underscore the rapid advancements in quantum computing. In 2022, a study published in Nature Physics[1] demonstrated coherent evolution through a quantum phase transition in a 2,000-qubit Ising chain, showcasing potential applications in quantum optimization, machine learning, and simulation tasks. Another significant achievement, published in Nature in the following year[2], realized quantum-critical spin-glass dynamics on a superconducting quantum annealer with thousands of qubits. This study demonstrated a clear advantage of QA over classical algorithms in simulating many-body quantum dynamics and solving complex optimization problems. These breakthroughs highlight the practical utility and scientific importance of D-Wave's quantum annealers in addressing intricate quantum phenomena and optimization challenges.

In the context of problem complexity, the dichotomy between **NP** (nondeterministic polynomial time) and **P** (polynomial time) stands as one of the most profound challenges in computer science and mathematics. **NP** encompasses problems whose solutions can be swiftly verified but not necessarily found efficiently, while **P** comprises problems for which both verification and solution discovery are feasible within reasonable timeframes[3–6].

The significance of distinguishing between **NP** and **P** lies in the fact that many real-world challenges fall within the **NP** or **NP-hard** categories[7–20]. Consequently, identifying optimal solutions for such problems often poses formidable computational hurdles. However, despite the complexity inherent in **NP-hard** problems, the pursuit

[1]Laboratory of Quantum Computing, Innopolis University, Innopolis 420500, Russia. [2]Research Center of the Artificial Intelligence Institute, Innopolis University, Innopolis 420500, Russia. [3]Center for Automation and Robotics, Innopolis University, Innopolis 420500, Russia. [4]Q Deep, Innopolis 420500, Russia. ✉email: h.salloum@innopolis.ru

of innovative computational methodologies remains imperative. Many problems either lack efficient solutions or possess suboptimal ones, necessitating continuous exploration of novel computational approaches.[21–24].

The QA technology provided by D-Wave Systems Inc., such as Quantum Processing Units (QPUs) and Hybrid Quantum Processing Units (HQPUs), is designed to solve complex combinatorial optimization problems[25–27]. The QPU is specifically designed to solve the Ising model, which is an NP-hard problem[28]. The Ising model is represented[29–31] as:

$$\mathrm{Obj}_{\mathrm{Ising}}(s, J, h) = \sum_{i} h_i s_i + \sum_{i<j} J_{ij} s_i s_j \tag{1}$$

where s_i represents the spin of qubit i, h_i is the local field bias, and J_{ij} is the coupling strength between qubits i and j.

This Ising model is equivalent to solving Quadratic Unconstrained Binary Optimization (QUBO) problems[32–34], represented as:

$$\mathrm{Obj}_{\mathrm{QUBO}}(x, Q) = x^T \cdot Q \cdot x \tag{2}$$

where x symbolizes a vector comprising binary variables, while Q is an $N \times N$ real-valued matrix delineating the interdependence among these variables. Each qubit in the QUBO problem corresponds to a variable in x, and the coefficients in Q represent the costs associated with qubit pairs. The QPU is an implementation of an undirected graph, with qubits as vertices and couplers as edges between them[35].

The breakthrough achieved by D-Wave with their hardware epitomizes the transformative potential of QA in revolutionizing problem-solving across diverse domains such as aerospace[36–38], cross-industry[39,40], energy[41–43], e-commerce[44], finance[45–47], telecommunication[48], and logistics[49]. By providing new avenues for addressing NP-hard problems, quantum technologies offer unprecedented opportunities to advance our understanding of complex systems and propel computational innovation to new heights. As researchers go deeper into harnessing the capabilities of quantum computing, the prospect of overcoming longstanding computational challenges becomes increasingly tangible. This heralds the dawn of a new era in computational science, marked by unprecedented innovation and discovery.

QA has enabled novel approaches for addressing urban traffic congestion by reformulating dynamic traffic signal control problems into a QUBO framework[50]. Subsequent research has extended these methods to the Traffic Assignment Problem by directly comparing quantum annealers, hybrid quantum-classical algorithms, and classical optimization techniques[51]. In addition, investigations into Virtual Traffic Lights and time-dependent traffic signal optimization have underscored the efficacy of quantum-enabled dynamic control systems in mitigating congestion and reducing delays[52,53]. Parallel developments in the realm of urban air mobility and network design[54,55], illustrate the versatility of quantum annealing in optimizing diverse transportation infrastructures and systems. Finally, the superiority of global control methods using quantum annealing over conventional simulated annealing in managing large-scale traffic systems reinforces the interconnection between these seminal works and the current study's focus on real-time traffic optimization through QUBO formulations[56].

In this regard, our work serves as a practical application that further demonstrates the computational capabilities of QA technology. More specifically, this study aims to address one of the most intricate challenges in modern transportation: traffic flow optimization. Our approach builds upon and enhances the foundational work undertaken by Volkswagen and D-Wave in 2017[57], refining their solution to improve its applicability in real-world scenarios. Our objective in this endeavor is to refine the solution, to be more applicable to real-world scenarios. Unlike previous work that treats entire traffic networks as single optimization problems, we exploit traffic flow locality by mini-scaling maps, dividing large urban areas into smaller sub-maps, each formulated as a separate QUBO problem. This ensures compatibility with D-Wave's Advantage QPU constraints while maintaining efficiency and scalability.

This paper addresses the **NP-hard** problem of traffic flow optimization, a critical challenge in modern logistics, by minimizing total congestion across all road segments. Traffic congestion, a widespread issue in urban centers worldwide, significantly intensifies this problem[58–60]. The 2023 INRIX Global Traffic Scorecard highlights the severity of congestion in major cities; New York City ranks as the most congested, with the average driver losing 101 hours annually to traffic delays. London, Mexico City, Paris, and Chicago follow closely, with drivers losing 99, 97, 96, and 96 hours each year, respectively[61–63].

The increase in traffic delays has resulted in substantial economic and social costs. In the United States, 40% of the top 25 urban areas studied in 2023 have met or exceeded pre-COVID levels of delay, compared to just 30% of all U.S. urban areas, while in Europe, two-thirds of cities have already met or exceeded pre-COVID levels of delay. Traffic congestion leads to a significant loss of time, which directly impacts productivity. In 2023, as shown in Fig. 1, the average U.S. driver lost 42 hours due to congestion, a 4-hour increase from 2022. In the UK, drivers lost 61 hours, marking a 4-hour increase, while in Germany, the average loss was 40 hours. The economic cost of congestion is staggering, with the U.S. economy losing over $70 billion, UK drivers incurring nearly £7.5 billion, and German drivers facing costs of 3.3 billion €. Traffic congestion presents numerous adverse effects, including significant time loss, economic costs, increased pollution, and health impacts. Addressing these challenges requires comprehensive strategies that encompass urban planning, transportation infrastructure improvements, and policy measures to manage demand and mitigate congestion.

Current approaches to address these issues include adaptive signal control systems[64] which dynamically adjust traffic lights based on real-time conditions to minimize wait times and improve flow. Route optimization

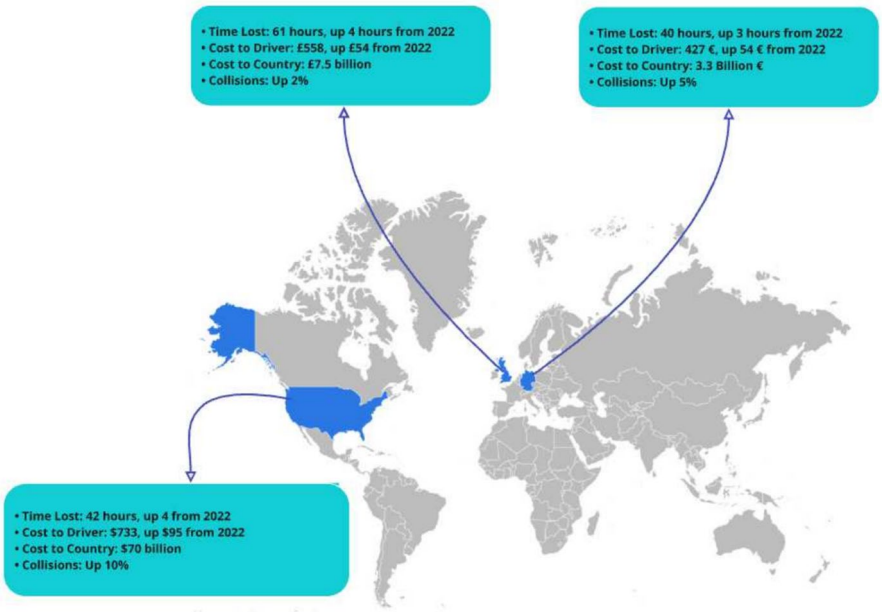


Fig. 1. Traffic congestion costs data for the United States, United Kingdom, and Germany according to 2023 INRIX Global Traffic Scorecard.

algorithms in navigation systems also help alleviate congestion by recommending less congested routes. Additionally, Intelligent Transportation Systems (ITS) are employed to manage demand and improve traffic efficiency. ITS integrates data-driven decision-making and advanced traffic monitoring technologies to optimize traffic flow, improve road safety, and reduce environmental impact[65]. As the world shifts toward autonomous vehicle technology, traffic flow optimization becomes increasingly critical for ensuring safe and efficient vehicle operation without human intervention[66,67]. This is where quantum computing becomes essential. Quantum computing can process large amounts of data simultaneously, making it significantly faster than traditional algorithms. This capability is essential for optimizing multiple factors at once, which is crucial for managing large-scale and dynamic traffic systems.

The motivation for this paper stems from the "Quantum Shuttle," the first real-world traffic navigation application utilizing quantum annealing, launched during the Web Summit 2019 in Lisbon, Portugal. Volkswagen AG partnered with the city of Lisbon to address the significant strain on the city's transit system caused by the influx of attendees at this large-scale conference[68]. The app was based on the work presented in[57], representing the first commercial application using a quantum processor to perform a critical live task. Our goal is to build upon[57], improving and extending their approach to develop a more advanced and practical real-world traffic optimization system based on quantum computing technologies.

In this work, we adopt a "Mini-scale" approach that decomposes large-scale traffic optimization problems into smaller, manageable subproblems as shown in Fig. 2. This strategy facilitates effective embedding and solution on D-Wave QPUs, overcoming the limitations of handling a monolithic QUBO formulation. Detailed discussions of this decomposition-highlighting the identification of critical congested road segments via Algorithm 1-are provided in section 2 (particularly section 2.3). This approach underscores that "Mini-scale" refers to the scale of the optimization problems addressed by the QPU, not a simplified model.

Methods

Quantum annealing in nutshell

QA is a quantum mechanical approach that can be considered a thermodynamic approach[69], using principles of quantum mechanics to solve combinatorial optimization problems, including NP-hard problems. While a quantum annealer operates as a quantum system, it additionally incorporates thermodynamic aspects due to its interaction with the surrounding environment and external control fields, functioning as a thermodynamic machine.

QA, categorized as a meta-heuristic tool, offers a promising avenue in this quest for efficient optimization[70–72]. Unlike classical algorithms, which rely on deterministic procedures, QA leverages quantum principles to explore vast solution spaces more effectively. By exploiting quantum tunneling, superposition, and entanglement, QA can navigate complex landscapes of possible solutions, often outperforming classical optimization methods in certain scenarios[73–75].

At the core of QA lies the time-dependent Schrödinger equation[76–78]:

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = \hat{H}(t)|\psi(t)\rangle \tag{3}$$

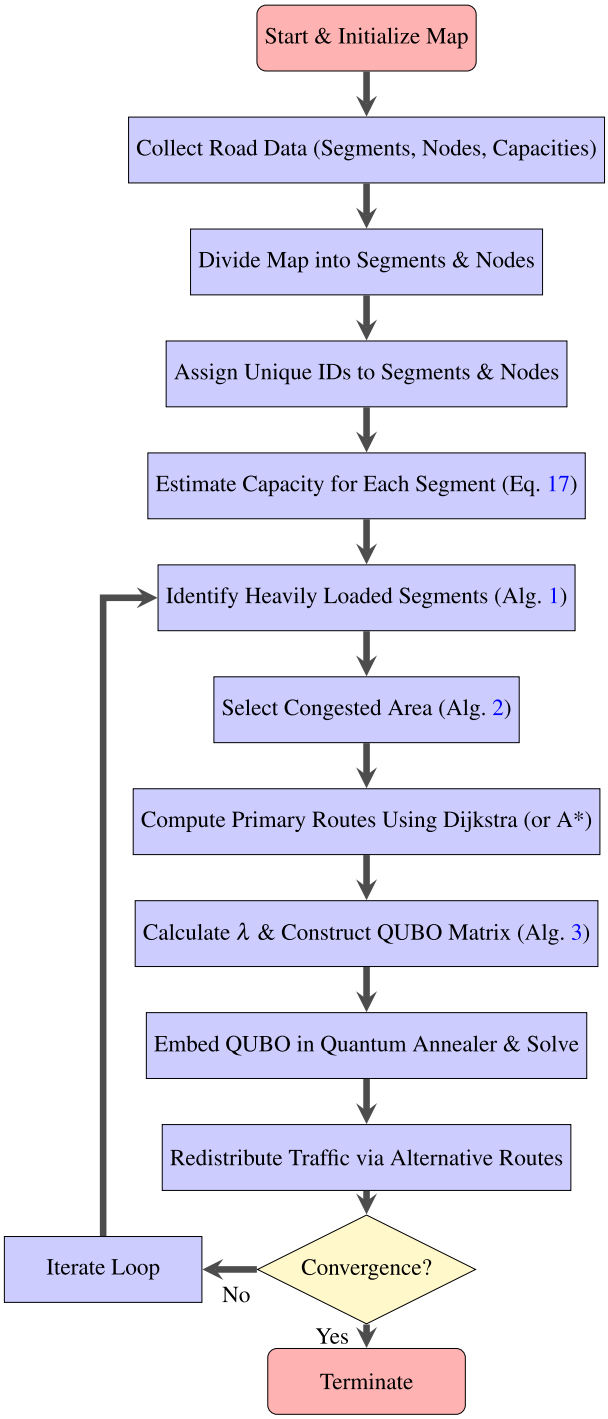


Fig. 2. Flowchart illustrating MTF to optimize traffic flow using QA.

Where:

$\hbar$ is the reduced Planck constant.
$|\psi(t)\rangle$ is the quantum state vector at time t.
$\hat{H}(t)$ is the time-dependent Hamiltonian.

The evolution of the quantum state vector $|\psi(t)\rangle$ over time is guided by the manipulation of the Hamiltonian $\hat{H}(t)$. During the annealing process, the Hamiltonian is transformed from an initial, easy-to-prepare form to the problem Hamiltonian, which encodes the problem's objective function. This gradual transformation ensures

that the system explores the solution space effectively, with higher probabilities assigned to states corresponding to lower energy levels, akin to finding optimal solutions.

A key advantage of QA is quantum tunneling, enabled by the inherent probabilistic nature of quantum mechanics. Quantum tunneling allows the system to traverse energy barriers, facilitating the escape from local minima and enabling the exploration of the entire solution space more efficiently.

The probability of tunneling through a barrier in QA can be described by the semiclassical WKB approximation[79,80]:

$$\text{tunnel} \propto e^{-\frac{2}{\hbar}\int \sqrt{2m(V(x)-E)}\,dx} \tag{4}$$

Where:

$$\begin{aligned}&\text{tunnel is the probability of tunneling through a barrier.}\\ &m \text{ is the mass of the particle.}\\ &V(x) \text{ is the potential energy.}\\ &E \text{ is the energy of the particle.}\end{aligned}$$

Moreover, the energy of the system at a given time t is computed using the time-dependent Hamiltonian[81,82]:

$$E(t) = \langle\psi(t)|\hat{H}(t)|\psi(t)\rangle \tag{5}$$

In the realm of quantum mechanics, an adiabatic process is characterized by the gradual alteration of a system's Hamiltonian, where the rate of change T_H is significantly slower than the characteristic time scale of the system's dynamics T_c, as stipulated by the condition $T_H \gg T_c$. This criterion ensures a smooth evolution of the system, allowing it to track changes in the Hamiltonian without inducing transitions between different energy levels.

The general solution to the time-dependent Schrödinger equation is expressed as a superposition of energy eigenstates:

$$\Psi(x,t) = \sum_n c_n \phi_n(x) \exp\left(-i\frac{E_n t}{\hbar}\right), \tag{6}$$

where $\Psi(x,t)$ represents the wave function of the quantum system at position x and time t, c_n are coefficients representing the contributions of each energy eigenstate, $\phi_n(x)$ are the spatial wave functions, and E_n are the corresponding energy eigenvalues.

The lowest energy that the system can reach is referred to as the ground state. For a quantum system undergoing an adiabatic process denoted by the transition $H_{\text{init}} \longrightarrow H_{\text{final}}$ with a sufficiently slow time scale T_H, the adiabatic theorem dictates that if the system starts in the n-th energy eigenstate of H_{init}, it will remain in the n-th energy eigenstate of H_{final} throughout the evolution, provided that the change is adiabatic[83].

The adiabatic theorem holds significant importance in quantum annealing (QA). Therefore, if a quantum system begins in its ground state $|\psi_0\rangle$ and evolves under a time-dependent Hamiltonian $H(t)$ at a sufficiently gradual pace, the system will remain in the instantaneous ground state $|\psi(t)\rangle$ throughout the evolution[73]:

$$|\psi(t)\rangle = |\psi_0\rangle \quad \text{for all } t. \tag{7}$$

This fundamental theorem ensures that QA commences with the system in a well-defined quantum state (the ground state of the initial Hamiltonian H_0) and maintains this state while gradually transitioning to the ground state of the problem Hamiltonian H_p as time progresses.

In Quantum Annealing (QA), quantum fluctuations are introduced via a transverse field, leading to non-commuting terms in the Hamiltonian. Specifically, in the D-Wave quantum annealer, the time-dependent Hamiltonian can be written[84] as:

$$H_{\text{QA}}(s) = -A(s)\sum_i \sigma_x^{(i)} + B(s)\left(\sum_i h_i \sigma_z^{(i)} + \sum_{i>j} J_{i,j}\sigma_z^{(i)}\sigma_z^{(j)}\right) \tag{8}$$

Here:

- $A(s)$ and $B(s)$ are smooth, monotonically varying functions of the annealing parameter $s \in [0,1]$, which control the relative contribution of quantum and classical effects.
- $\sigma_x^{(i)}$ and $\sigma_z^{(i)}$ are Pauli matrices acting on the i-th qubit, corresponding to the transverse and longitudinal directions, respectively.
- h_i represents the local magnetic field acting on the i-th qubit.
- $J_{i,j}$ denotes the coupling strength between qubits i and j.

The Hamiltonian can be decomposed into two parts:

- The **Initial Hamiltonian** H_{initial} governs the quantum fluctuation term:

$$H_{\text{initial}} = -A(s) \sum_i \sigma_x^{(i)} \tag{9}$$

- The **Problem Hamiltonian** H_p encodes the classical Ising model problem:

$$H_p = B(s) \left(\sum_i h_i \sigma_z^{(i)} + \sum_{i>j} J_{i,j} \sigma_z^{(i)} \sigma_z^{(j)} \right) \tag{10}$$

The QA process evolves over a time interval from $t = 0$ to $t = t_f$, which is regulated by the user through the **annealing time** parameter or customized by defining a schedule via **anneal schedule**. The normalized annealing parameter $s \in [0, 1]$ is used to parameterize the evolution. At the initial moment ($t = 0$ or $s = 0$), the quantum Hamiltonian is dominated by the transverse field, such that $A(0) \gg B(0)$, placing the system in a quantum ground state where each spin s_i is in a superposition of classical states $s_i = \pm 1$. This corresponds to a highly quantum regime. As time progresses, $A(s)$ gradually decreases, while $B(s)$ increases. By the end of the annealing process ($t = t_f$ or $s = 1$), the system transitions to a classical regime where $A(1) \ll B(1)$. At this stage, the qubits behave as classical spin variables, and the Pauli matrices $\sigma_z^{(i)}$ are replaced by classical Ising spins $s_i = \pm 1$. Thus, the final Hamiltonian describes a classical Ising spin model in equation 1.

The QA process aims to find the ground state of H_p, which corresponds to the optimal solution of the optimization problem. Thus, the process starts with the known ground state of an easily prepared initial Hamiltonian H_0 and slowly evolves the system through a time-varying Hamiltonian $H(t)$ according to the adiabatic theorem. As t increases, the system adiabatically reaches the ground state of H_p, providing the solution to the optimization problem.

Hence, QA can be perceived as a specialized variant of the classical Simulated Annealing algorithm. Its primary focus is on tackling optimization problems that involve real-valued functions of Boolean variables. Integrating QA into **NP** and **NP-Hard** problem-solving offers an alternative perspective on addressing the inherent complexity of these problems. By harnessing quantum principles, QA algorithms can efficiently explore large solution spaces, potentially overcoming the computational limitations of classical methods and offering new insights into hard problem-solving strategies.

Formulation of the traffic congestion cost function

This study employs QA techniques to optimize traffic flow, drawing upon the foundational work of Neukart et al. on traffic optimization[57]. Our objective is to minimize the overall cost associated with road segments while efficiently assigning alternative routes to vehicles. Unlike previous works that consider a single starting point and destination, our approach accounts for multiple starting points and destinations. We endeavor to embed the problem directly onto the QPU, which promises substantial speedup and a reduction in problem size.

The traffic flow optimization problem, as formulated in classical works, often hinges on the principles of flow conservation, capacity constraints, and an objective function aimed at improving overall system performance. In order to construct cost function that represents the cost of traffic congestion, we start from established formulations and then adapt them to the QUBO framework, which is suitable for a quantum annealer due to its equivalence to the Ising model. The objective function depends on the desired goal, such as minimizing total travel time or congestion. However, similar to the assumptions used in[89], we assume that the time required to traverse a road segment is proportional to a function of the number of vehicles occupying it . Thus, the cost function for a road segment s is defined as follows:

$$\text{cost}(s) = \left(\sum_{q_{ij} \in B_s} w_{ij} q_{ij} \right)^2 \tag{11}$$

Here:

- q_{ij} represents the presence of vehicle i on segment j, where $q_{ij} = 1$ if the vehicle traverses segment j and $q_{ij} = 0$ otherwise.
- B_s denotes the set of vehicles traversing segment s.
- $w_{ij} = w_s$ is the weight assigned to segment s, reflecting the congestion weight.

To ensure each vehicle is assigned exactly one route, we introduce the following constraint:

$$\left(\sum_{j=1}^{3} q_{ij} \right)^2 - 2 \sum_{j=1}^{3} q_{ij} + 1 = 0 \tag{12}$$

This constraint guarantees that each vehicle i selects precisely one of the three available routes. The first term on the left-hand side sums up the presence of vehicle i on each route, while the second term enforces the exclusion of scenarios where a vehicle simultaneously takes multiple routes. Hence, our global objective function for the QUBO problem is revised as follows:

$$\mathcal{O} = \sum_{s \in S} \left(\sum_{q_{ij} \in B_s} w_{ij} q_{ij} \right)^2 + \lambda \sum_i \left(\sum_j q_{ij} - 1 \right)^2 \tag{13}$$

Here:

- S represents the set of all street segments.
- λ is a scaling parameter ensuring compliance with the route assignment constraint for each vehicle (calculated as in Algorithm 3).

The second term penalizes scenarios where a vehicle takes more than one route. This assumption ensures that vehicles are distributed across routes in a way that minimizes congestion while avoiding the complexity of modeling flow balance across nodes. The result is an optimization problem where the main focus is on the binary decision-making process for vehicle assignment, rather than flow conservation. This method can simplify the formulation while still addressing the key challenge of minimizing congestion across road segments. To enhance traffic management optimization, we introduce a novel approach aimed at circumventing the complexities associated with solving a single, extensive QUBO problem. Rather than addressing the problem as a monolithic entity, we advocate for its decomposition into smaller, more tractable subproblems, thereby mitigating overall complexity. This approach will render traffic flow optimization more feasible on QPUs, as embedding a large QUBO into the QPU can become infeasible. Thus, this work allows for utilizing D-Wave QPUs, not just hybrid QPUs.

Our strategy recognizes that traffic congestion correlates directly with the volume of vehicles traversing roadways. We decouple the optimization process from temporal constraints by focusing on identifying optimal routes based solely on their starting and ending points. Each route is treated as an independent entity, unaffected by temporal fluctuations, thereby fostering a robust solution framework that promotes efficient traffic flow and ensures consistency across diverse traffic scenarios.

Moreover, our approach incorporates the methods outlined in[57] to assign vehicles to alternative routes. Rather than attempting to manage all vehicles simultaneously, we concentrate on the most congested areas. By identifying high-traffic segments, we focus our efforts on analyzing and optimizing small, localized regions experiencing the highest levels of congestion. This targeted approach leverages the principle of continuity, wherein adjacent segments along the same roadway tend to exhibit similar congestion levels due to the consistent flow of vehicles entering and exiting nodes.

We propose a new cost function for the selected congested areas, formulated as follows:

$$\mathcal{C} = \sum_{s \in S} \left(\sum_{q_{ij} \in B_s} w_{ij}^2 + \sum_{s' \in S_1} h_{s'}^2 \right) q_{ij} + \lambda \sum_i \left(\sum_j q_{ij} - 1 \right)^2 \tag{14}$$

where S_1 represents the set of intersections external to the selected congested area, and $h_{s'}$ denotes the congestion weights of these external intersections, influencing the redistribution of vehicular traffic within the designated zone. Specifically, if we denote the selected area by S_{area}, then $h_s = \omega_s$ for $s \notin S_{\text{area}}$. The intuition behind this cost function is to ensure that solving congestion in one area does not create additional congestion elsewhere. The added weights help prevent congestion from being transferred to other regions, and to get benefit from the QPU sped up, the size of the problem is constrained to fit the Pegasus hardware, which can embed large problems utilizing up to 177 logical qubits[90].

MTF computational processes

The map preprocessing for traffic flow optimization starts with collecting data, including road segments, node positions (intersections), and road capacities. Then, we divide the map into segments and nodes, assigning a unique ID to each. Additionally, for each pair of nodes, different IDs are assigned for each direction. As a result, the road network is modeled as an undirected graph $G = (V, S)$, where V represents nodes corresponding to intersections or key decision points, and S represents edges that model the road segments between these nodes. The network segmentation divides the road between intersections as edges and treats the intersections themselves as nodes. Capacity estimation for each edge is an important step, as it determines the constraints used in the optimization problem, helping ensure that the traffic flow on each road segment does not exceed its capacity. This will be reflected in Eq. 17.

The process of constructing the primary route for new users or those deviating from previous routes begins with route computation. When it comes to routing optimization, classical algorithms for finding the shortest path include Dijkstra's algorithm, the Bellman-Ford algorithm, and A*. Dijkstra's algorithm is especially advantageous in the context of road networks with non-negative edge weights because it efficiently finds the shortest path from a source node to all other nodes in $O(|S| + |V|log|V|)$ time, using a priority queue for efficient updates. Bellman-Ford, while more flexible for handling negative weights, operates in $O(|V|.|E|)$ time, making it slower for large networks[92]. Dijkstra's algorithm has been evaluated in comparison to A* and dynamic programming approaches for analyzing vehicle routes in urban traffic, particularly with regard to optimization accuracy. Studies show that Dijkstra maintains a lower average error rate in certain conditions, though A* can outperform it when an effective heuristic is applied. Dynamic programming, while robust, can become computationally intensive, making Dijkstra more suitable for real-time traffic pathfinding due to its balance of

simplicity and efficiency in handling real-world traffic conditions where edge weights (representing travel times) are non-negative and can change frequently depending on road conditions[93].

Once we assign the initial routes to each car in the network, by using Algorithm 1, our goal is to find the most congested road segment. This helps to focus on areas that need optimization to improve traffic flow.

```
function initialize_map():
segments, nodes ← divide_map()
for each segment in segments do
    assign unique id(segment)
end for
for each pair of nodes (i, j) in nodes do
    assign unique id for each direction(i, j)
end for
function identify heavily loaded segments(segments):
for each segment in segments do
    n_ij ← compute highest possible car count(segment)
    segment.product value ← calculate product value(segment, n_ij)
end for
most congested segment ← None
max product value ← −1
for each segment in segments do
    if segment.product value > max product value then
        max product value ← segment.product value
        most congested segment ← segment
    end if
end for
return most congested segment
```

Algorithm 1. Initialization and identification of heavily loaded segments

Each road segment has several factors that tell us how busy it is. One important factor is the number of cars that can be on the segment at a given time, which we will call n_j (the maximum number of cars on segment *j*). In addition to the number of cars, each car has a certain "weight" based on its impact on congestion (for example, larger vehicles like trucks might have a higher weight than smaller cars). We'll call this $w_{i,j}$, which represents the weight of car *i* on segment *j*.

To determine which road segment is the most congested, we calculate the product of the maximum number of cars (n_j) and the weight of each car on that segment. The segment with the highest product is the most congested one at that moment.

$$S_j = n_j * w_{i,j} \tag{15}$$

Initialization:
Initialize network variables: nodes, edges.
Set parameters: number_of_alternative_routes, max_iterations.
segments ← {}
selected_edges ← {}
routes ← {}

Clearing existing data:
segments.clear()
selected_edges.clear()
routes.clear()

Identify Most Congested Segment:
most congested segment ← identify heavily loaded segments(segments)
current_edge ← most congested segment

Iterative Area Selection:
for iteration = 1 **to** max_iterations **do**
  Expand Area:
  Extract original routes and corresponding weights.
  Identify alternative routes similar to the original ones.
  Construct a sub-map considering the congested segment and alternative routes.
  Compute the average congestion S_R using Equation 16.
  Select new segments maintaining the same start and end intersections.
  Add newly selected edges to selected_edges.
  Add newly selected segments to segments.
  Update Current Edge:
  Determine next edge based on expansion criteria:
  Select the next edge where S_j satisfies Equation 17.
  Assign current_edge ← next selected edge.
end for

Finalization:
Unmark selected edges in the network.
Retrieve and store list of alternative routes based on selected segments.
return routes

Algorithm 2. Area selection

For area selection, we implement Algorithm 2 to represent the congestion of a segment by encompassing a set of roads as in the equation 16.

$$S_R = \frac{\sum_{j \in R} S_j}{|R|} \tag{16}$$

Here, S_R represents the average congestion of the segment of a route. Subsequently, we extract all traffic participants with routes traversing this road. Then, we determine alternative routes for this segment and, connect each user to these alternative routes. This process repeats until the average load of the alternative routes exceeds the load of the redistributed segment.

$$S_j \leq \frac{\sum_{R \in Paths} S_R}{|Paths|} \tag{17}$$

Each alternative path must initiate from the same intersections as the original road segment. Consequently, a small area is delineated for conversion into a matrix form, facilitating the redistribution of users within this territory via QUBO matrix transformations for efficient processing.

To construct the Q matrix for our traffic optimization problem, we proceed as follows:

1. **Initialization**:

$$\mathrm{Q} = 0_{N \times N} \tag{18}$$

where N represents the total number of alternative routes for all vehicles.

2. **Segment Congestion**: When two routes j and j' share a street segment s:
   (a) Add w_{ij}^2 to the diagonal entry at index $I(q_{ij})$ for each vehicle i with route j containing segment s.
   (b) Add $2w_{ij}$ to the off-diagonal element at indices $I(q_{i_1 j})$ and $I(q_{i_2 j})$ for every pair of vehicles i_1 and i_2 taking route j containing segment s.

3. **Route Uniqueness Constraint**:
   (a) For each vehicle i with possible route j, add $-\lambda$ to the diagonal entry of Q at index $I(q_{ij})$.
   (b) For each cross-term arising from the segment congestion term, add 2λ to the corresponding off-diagonal element.

4. **External Intersection Congestion**: For each vehicle i with possible route j, add h_s^2 to the diagonal entry of Q at index $I(q_{ij})$.

To summarize, the updates to the Q matrix are as follows:

$$\begin{cases} \mathrm{Q}_{ii} = (w_{ij}^2 - \lambda + h_s^2) : i = j \\ \mathrm{Q}_{ij} = (2w_{ij} + 2\lambda) : i \neq j \end{cases} \tag{19}$$

As mentioned earlier, due to the size of the problem, the Q matrix is constrained to fit the Pegasus hardware, which can embed large problems utilizing up to 177 logical qubits.

These steps ensure a comprehensive and sophisticated approach to constructing the Q matrix for optimizing traffic flow. The process accounts for segment congestion, enforces unique routes for each vehicle, and incorporates the impact of external intersection congestion. We represent λ as the Maximum Intersection Cost. Algorithm 3 demonstrates how λ was calculated and how the Q matrix is constructed.

Require: $\mathscr{S}$: Set of road segments, $\mathscr{A}$: Alternative routes per vehicle
Ensure: **Q**: QUBO matrix

Function ComputeEdges.$\mathscr{S}$
$\mathscr{E} \leftarrow \emptyset$ Initialize edge set
for $s \in \mathscr{S}$ **do**
  for $s' \in \mathscr{S}, s' \neq s$ **do**
    if s and s' share a common node (intersection) **then**
      $\mathscr{E} \leftarrow \mathscr{E} \cup \{(s, s')\}$
    end if
  end for
end for
Return $\mathscr{E}$
Function ComputeIntersectionsv_1, v_2
$C \leftarrow 0$
for $a_1 \in \mathscr{A}(v_1), a_2 \in \mathscr{A}(v_2)$ **do**
  $C \leftarrow C + \text{IntersectionCount}(v_1[0], a_1, v_2[0], a_2)$
end for
return C
Function ComputePenaltyLambda
$\lambda \leftarrow 0$
for $v_1 \in \mathscr{A}$ **do**
  $C_{\text{sum}} \leftarrow \sum_{v_2 \neq v_1} \text{ComputeIntersections}(v_1, v_2)$
  $\lambda \leftarrow \max(\lambda, C_{\text{sum}})$
end for
return λ
Function ConstructQUBO.$\mathscr{S}, \mathscr{A}$
$\mathscr{E} \leftarrow \text{ComputeEdges}(\mathscr{S})$
if $\lambda = 0$ **then**
  $\lambda \leftarrow \text{ComputePenaltyLambda}()$
end if
$N \leftarrow |\mathscr{A}|$, $\mathbf{Q} \leftarrow 0_{N \times N}$
for $v_i \in \mathscr{A}, a_k \in \mathscr{A}(v_i)$ **do**
  $id \leftarrow N \cdot i + k$
  $\mathbf{Q}[id, id] \leftarrow -\lambda + \text{IntersectionCount}(\mathscr{E}, \mathscr{S})^2$
  for $k' > k$ and $k' \mod N \neq 0$ **do**
    $\mathbf{Q}[id, k'] \leftarrow \mathbf{Q}[id, k'] + 2\lambda$
  end for
end for
for $(v_i, v_j) \in \mathscr{A}, i < j$ **do**
  for $a_k \in \mathscr{A}(v_i), a_m \in \mathscr{A}(v_j)$ **do**
    $id_1, id_2 \leftarrow N \cdot i + k, N \cdot j + m$
    if $id_1 \geq id_2$ **then**
      continue
    end if
    $V \leftarrow \text{IntersectionCount}(v_i[0], a_k, v_j[0], a_m)$
    $\mathbf{Q}[id_1, id_2] \leftarrow \mathbf{Q}[id_1, id_2] + 2V$
    $\mathbf{Q}[id_1, id_1], \mathbf{Q}[id_2, id_2] \leftarrow \mathbf{Q}[id_1, id_1] + V, \mathbf{Q}[id_2, id_2] + V$
  end for
end for
return **Q**
$\mathbf{Q} = \text{ConstructQUBO}(\mathscr{S}, \mathscr{A})$

Algorithm 3. QUBO constructor

The following table provides a detailed explanation of all variables used in the QUBO Constructor for traffic flow optimization.

Variable	Description
$\mathcal{S}$	Set of road segments, where each segment contains available alternative routes and traffic flow properties.
$\mathcal{A}$	Set of alternative routes available for each vehicle, representing possible paths in the traffic network.
$\mathcal{E}$	Set of road edges, constructed from $\mathcal{S}$, representing the connections between different road segments. This is used to determine intersection conflicts. It is obtained by analyzing which segments are adjacent and detecting common nodes in the traffic network graph.
v_1, v_2	Identifiers for two vehicles in the road network whose routes are analyzed for intersections.

Variable	Description
a_1, a_2	Alternative routes associated with vehicles v_1 and v_2, defining possible paths they can take.
s_1, s_2	Road segments corresponding to vehicles v_1 and v_2, where potential intersections are calculated.
C	Counter that accumulates the number of route intersections between vehicles.
$\text{IntersectionCount}(s_1, a_1, s_2, a_2)$	Function that computes the number of intersection conflicts between two given alternative routes on road segments s_1 and s_2.
λ	Penalty parameter representing the worst-case intersection scenario. It is dynamically computed to ensure congestion avoidance.
C_{sum}	Total number of intersection conflicts experienced by a vehicle across all other vehicles in the system.
$\max(\lambda, C_{\text{sum}})$	Function that updates λ to ensure it reflects the worst-case congestion penalty in the network.
N	Total number of alternative routes across all vehicles, used to determine the dimensions of the QUBO matrix.
Q	The QUBO matrix, where each element $Q[i, j]$ represents a cost function term for selecting or avoiding a specific route.
v_i, v_j	Vehicles indexed by i and j, whose alternative routes are analyzed for conflicts.
a_k, a_m	Alternative routes selected for vehicles v_i and v_j.
id	Unique index representing a vehicle-route combination in the QUBO matrix.
r	Alternative route object associated with a specific vehicle and segment, used to determine intersections.
$Q[id, id]$	Represents the diagonal element in the QUBO matrix, applying penalties for intersection-heavy routes.
$Q[id_1, id_2]$	Represents off-diagonal elements in the QUBO matrix, penalizing the selection of conflicting routes.
V	Intersection penalty computed for conflicting alternative routes.

Finally, To implement the MTF Algorithm, as outlined in Algorithm 4, we utilize Algorithms 1, 2, and 3. After constructing the Q Matrix, it is embedded into a D-Wave Quantum Annealer for quantum computation. This phase entails executing QA on the Q Matrix to ascertain an optimized traffic flow solution denoted as $\mathcal{F}^*$, followed by a meticulous redistribution of traffic resources. The iterative refinement process persists until convergence is attained, ascertained by criteria such as minimal perturbations in traffic flow or the attainment of specified optimization benchmarks. The Converged function assumes a pivotal role in this iterative process by meticulously evaluating these criteria and signaling convergence with a return of true, or continuing optimization otherwise with a return of false. The procedure for creating routes and maps (refer to Fig. 3) can be summarized as follows:

- The area requiring optimization is selected, and all vehicle paths are extracted, as depicted in Fig. 3a.
- The most congested segment is identified using Equation 15, as illustrated in Fig. 3b.
- The vehicle paths that are trailing the identified segment are selected, as shown in Fig. 3c.
- Based on the previous results, new segments are added to the most congested segment using Equation 17, as illustrated in Fig. 3d.
- Finally, new routes are constructed, and the map is updated accordingly, as shown in Fig. 3e. The resultant data will be prepared for QUBO formulation and subsequently sent to Leap Cloud for execution on the D-Wave quantum annealer.

Here we should note that the map of Almaty is rich with routes; the connectivity is very good as shown in Fig. 3f.

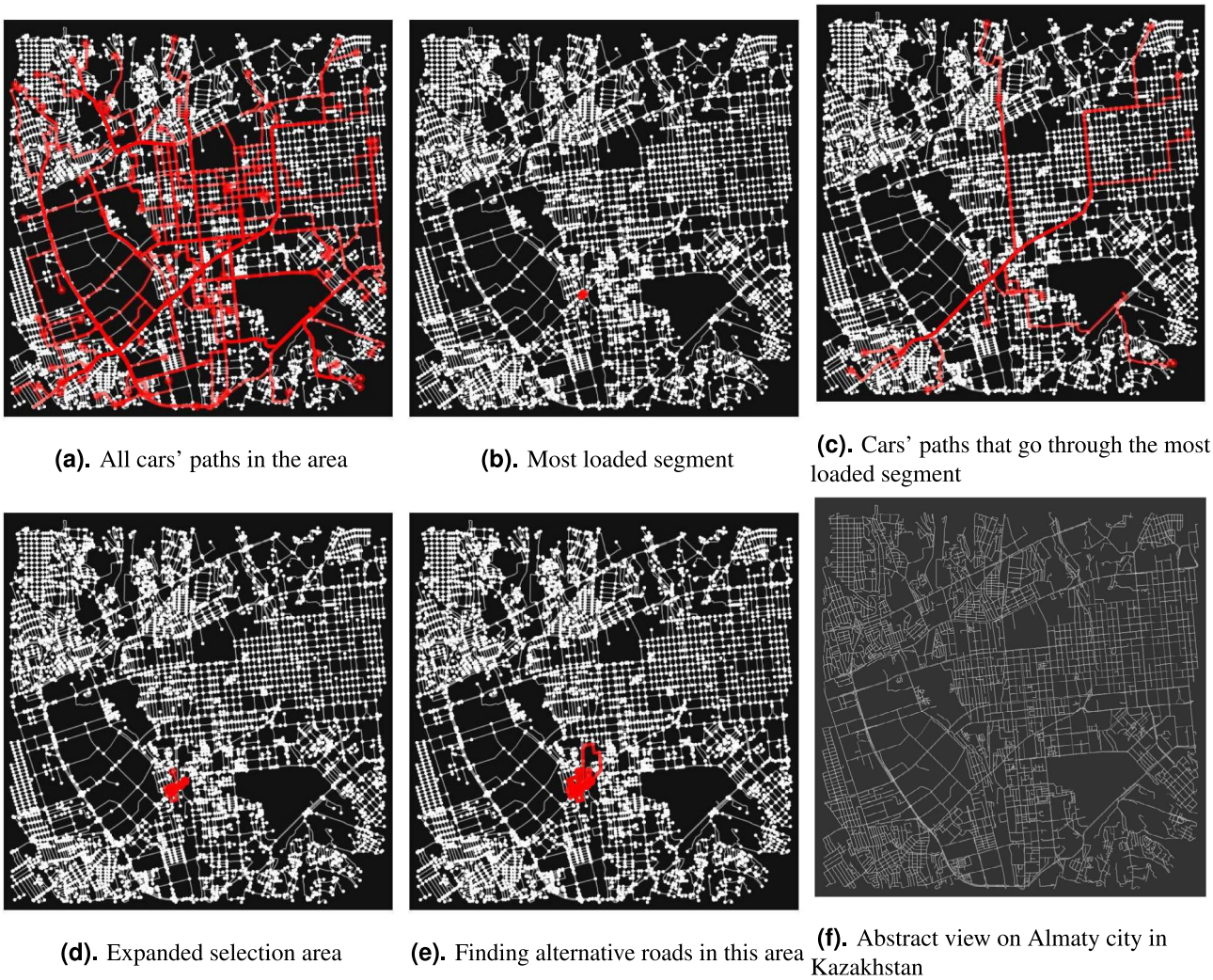
Results

A series of experiments were conducted in Almaty, Kazakhstan, using datasets of varying sizes: 100, 200, 300, 400, and 500 cars. We applied weights similarly as described in[89] to compare with our work and the work[57]. As mentioned earlier, the experimental setup allowed cars to have multiple start and end points, simulating diverse traffic scenarios. The initial solution for the experiments was created using the shortest path algorithm in OpenStreetMap (OSMnx)[91].

The results demonstrated a significant reduction in traffic congestion when using the MTF method. Specifically, the MTF approach optimized traffic flow by efficiently reallocating routes, thus minimizing congestion at critical intersections and along major routes. In contrast, the other method exhibited minimal improvement in traffic conditions, with some scenarios showing negligible or no improvement at all. This can be attributed to the limitations in dynamically adjusting to varying traffic patterns and its less effective handling of route weights.

Figure 4 presents an analysis of traffic congestion improvements across different car datasets. It illustrates the distribution of congestion-reduction enhancements using the MTF method across multiple runs, with each box plot representing different initial route assignments. In this context, the "Median Congestion Reduction" (MCR) refers to the median percentage decrease in segment occupancy (i.e., vehicle density) relative to a baseline scenario, calculated over multiple simulations. By focusing on the median rather than the mean, MCR mitigates the impact of outliers, thereby providing a more robust and representative assessment of how effectively each approach alleviates typical congestion levels.

As shown, the median improvement (MCR) increases steadily from approximately 25% for 100 cars to over 60% for 500 cars. For instance, in the 500-car dataset, the MTF method reduced congestion by approximately



(a). All cars' paths in the area

(b). Most loaded segment

(c). Cars' paths that go through the most loaded segment

(d). Expanded selection area

(e). Finding alternative roads in this area

(f). Abstract view on Almaty city in Kazakhstan

Fig. 3. Procedure for creating routes and maps.

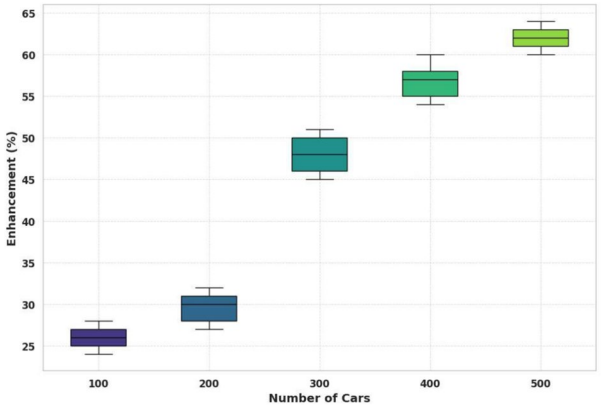


Fig. 4. Traffic Enhancement by Number of Cars.

62% as shown in Fig. 5. Building on this performance, Fig. 6 provides a heatmap visualization for the 500-car dataset, illustrating the possible maximum traffic density across various road segments. The heatmap for the MTF method shows significant results in optimizing congestion with considerably fewer high-density areas. This highlights the MTF method's effectiveness in preventing traffic bottlenecks and ensuring smoother traffic flow.

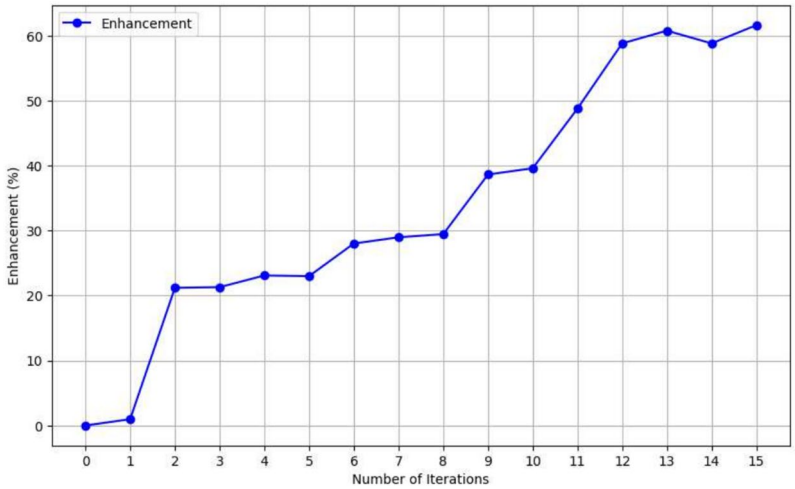
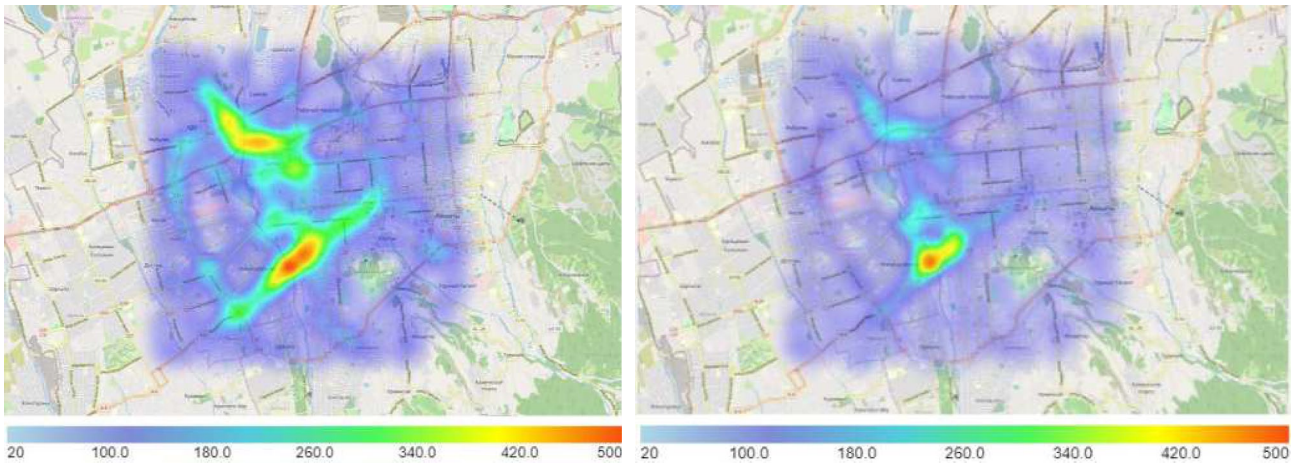


Fig. 5. Enhancement vs Number of Iterations for MTF Algorithm for the 500-car experminet.



(a). Traffic congestion before applying MTF **(b).** Traffic congestion after applying MTF

Fig. 6. Heatmap for the density of cars over kilometers for the 500-car experiment.

Dataset Size (Cars)	MCR (MTF)	MCR (Neukart et al.[57] + Tambunan et al.[89])	WRS p-Value
100	25%	14%	< 0.01
200	30%	12%	< 0.01
300	50%	10%	< 0.01
400	55%	9%	< 0.01
500	62%	7%	< 0.01

Table 1. Comparative Results on Median Congestion Reduction (MCR) between (MTF) and (Neukrat et al.[57] with weights added[89]).

Computational time is a critical factor in evaluating the practicality of traffic optimization algorithms. The previous algorithm's Q matrix is exceptionally large, making it impractical to embed directly in the QPU. Consequently, we adopted a hybrid quantum-classical approach, resulting in a computation time of approximately 3 seconds. In contrast, our implementation of the MTF method exhibited superior computational efficiency. The total computation time required for the MTF method to reach convergence across all experiments ranged between approximately 0.150 and 0.225 seconds, as each complete optimization run typically required under a second. While the method involved multiple internal iterations (e.g., 12 iterations in the 500-car case), the total time remained within this range due to the QPU's rapid execution-each iteration operating at a millisecond

timescale over 1000 readings, as reported by D-Wave[94]. The marked decrease in computational time is a result of embedding directly into the Pegasus topology, facilitated by the streamlined algorithmic structure of the MTF method and its adept management of smaller, more manageable Q matrices. To further substantiate the efficacy of the MTF method, we conducted Wilcoxon rank-sum tests (WRS)[95] on the congestion reduction results for each dataset size (100, 200, 300, 400, and 500 cars) as shown in Table 1.

In every case, the p-values fall below the 0.01 threshold, thus rejecting the null hypothesis of no difference between the MTF method and the alternative. Consequently, these statistical tests confirm that the MTF method provides a greater reduction in traffic congestion and further support the suitability of the MTF approach for large-scale, real-time traffic optimization.

Input: Traffic network $\mathscr{T}$, maximum iterations $I_{\max}$, number of alternative routes R
Output: Optimized traffic flow $\mathscr{F}^*$

Initialize network: nodes, edges, and assign unique IDs.
Set iteration counter $I \leftarrow 0$.

while $I < I_{\max}$ **and** not Converged() **do**
  Run Algorithm 1: Identify heavily loaded segments.
  Run Algorithm 2: Select the congested area.
  Compute λ using the current traffic intersection costs (see Algorithm 3).
  Run Algorithm 3: Construct the QUBO matrix $\mathbf{Q}$.
  Ensure that the QUBO matrix size satisfies $N \leq 177$ logical qubits.
  Embed $\mathbf{Q}$ into the D-Wave Quantum Annealer.
  $\mathscr{F}^* \leftarrow$ Execute Quantum Annealing on $\mathbf{Q}$.
  Redistribute traffic according to $\mathscr{F}^*$.
  Update the traffic network $\mathscr{T}$ with the new vehicle assignments.
  Increment iteration counter $I \leftarrow I + 1$.
end while

Function: Converged()
  Evaluate stopping criteria (e.g., change in congestion levels $< \varepsilon$).
  return `true` if converged, `false` otherwise.

Algorithm 4. MTF Algorithm

It is important to emphasize that these results that we obtained are exploratory in nature. They serve as preliminary evidence of the potential effectiveness of the MTF method in diverse traffic conditions, and further investigation is warranted to substantiate these findings.

Limitation and future work

While our experiments have shown promising results, several limitations must be acknowledged. One of the primary considerations in our study is the decision to compare our method with an existing quantum annealing approach rather than traditional classical optimization techniques. Classical methods were not considered in our comparative analysis because our research is specifically focused on advancing quantum annealing techniques for traffic optimization. Classical algorithms, such as metaheuristic or exact solvers, are highly effective for small-scale traffic networks but struggle significantly as the problem scale increases. Traffic optimization is an NP-hard combinatorial problem, and the solution space grows exponentially as the number of vehicles and intersections increases. In large urban networks, even the most advanced classical solvers encounter severe computational bottlenecks, making real-time optimization impractical. Quantum annealing, in contrast, offers a fundamentally different computational paradigm which has demonstrated promising scalability trends particularly for NP-hard QUBO formulations[96]. This allows for effective congestion reduction without the dramatic computational spikes typically observed in classical methods. We also acknowledge the current constraints of quantum annealing hardware, including limited qubit connectivity, susceptibility to noise, and constraints on QUBO problem sizes. Although today's quantum hardware does not yet surpass classical solvers in all cases, it is rapidly evolving. Future generations of quantum annealers will feature enhanced qubit capacity and connectivity, making larger traffic optimization problems feasible. Our study provides a methodological foundation for quantum traffic optimization, ensuring that as quantum hardware improves, our approach can readily exploit its capabilities.

Another limitation of our study is the restricted scope of data. Our experiments were conducted within a specific map in Almaty, Kazakhstan, which may not fully capture the diversity of traffic conditions found in larger urban environments. Additionally, our experiments did not account for certain real-world constraints, such as traffic light synchronization, road capacity limits, and environmental conditions. These factors can significantly influence traffic flow and should be incorporated into future studies to enhance the model's realism.

Furthermore, our current implementation faces potential scalability limitations due to the computational demands of large QUBO matrices, which may become impractical for direct embedding onto QPUs.

To address these challenges, we plan to explore dividing maps into submaps (A-B decomposition) and embedding them onto D-Wave's Advantage 2 quantum annealer, which utilizes the Zephyr topology[87]. This architecture is designed to enhance scalability by supporting larger QUBO matrices while maintaining computational efficiency. Additionally, we aim to incorporate real-world traffic constraints, such as traffic signals, road capacities, weather variations, and driver behavior, to improve the applicability and realism of our method. By addressing these limitations, we aim to further advance quantum traffic optimization toward city-scale implementations, ensuring greater effectiveness in real-world traffic management scenarios.

Conclusion

The findings in Table 1 demonstrate that the MTF method not only yields a higher-quality solution-as measured by substantial congestion reductions-but also provides computational gains. In particular, for the 500-car dataset, MTF reduces congestion by approximately 62%, reflecting its effectiveness in preventing traffic bottlenecks and ensuring smoother traffic flow. Furthermore, MTF maintains low run times (0.150-0.225 seconds), underscoring its computational efficiency. However, we note that while MTF excels in more complex road networks, simpler maps might benefit from the[57] approach. Beyond technical performance, real-world benefits of reduced congestion include lower fuel consumption, diminished economic losses due to travel delays, and decreased CO_2 emissions. By addressing both solution quality and computational speed, MTF offers a promising avenue for traffic management, appealing not only to researchers but also to policymakers seeking to improve transportation efficiency and minimize environmental impact.

Data availability

The datasets generated and/or analyzed in this study involve specialized traffic optimization data with potential security and privacy implications. Consequently, public access is restricted. However, researchers may request access to the data from the corresponding author, Hadi Salloum, and such requests will be reviewed individually to ensure appropriate and secure data usage.

Received: 25 October 2024; Accepted: 27 May 2025
Published online: 02 July 2025

References

1. King, A. D. et al. Coherent quantum annealing in a programmable 2,000 qubit Ising chain. *Nat. Phys.* **18**(11), 1324–1328 (2022).
2. King, A. D. et al. Quantum critical dynamics in a 5,000-qubit programmable spin glass. *Nature* **617**(7959), 61–66 (2023).
3. Cook, S. The P versus NP problem. *Clay Math. Inst.* **2**, 6 (2000).
4. Aaronson, S. (2016). P =? NP. Open problems in mathematics. Springer, pp. 1–122.
5. Sudan, M. (2010). The P vs. NP problem.
6. Wigderson, A. *Mathematics and computation: A theory revolutionizing technology and science* (Princeton University Press, Princeton, 2019).
7. Castaneda, J., Ghorbani, E., Ammouriova, M., Panadero, J. & Juan, A. A. Optimizing transport logistics under uncertainty with simheuristics: Concepts, review and trends. *Logistics* **6**(3), 42 (2022) (**MDPI**).
8. Juan, A. A., Kelton, W. D., Currie, C. S., & Faulin, J. (2018). Simheuristics applications: dealing with uncertainty in logistics, transportation, and other supply chain areas. In *2018 winter simulation conference (WSC)* (pp. 3048–3059). IEEE.
9. Tindell, K. W., Burns, A. & Wellings, A. J. Allocating hard real-time tasks: an NP-hard problem made easy. *Real-Time Syst.* **4**(2), 145–165 (1992).
10. Haider, A., Potter, R., & Nakao, A. (2009). Challenges in resource allocation in network virtualization. In *20th ITC specialist seminar* (Vol. 18, No. 2009). ITC.
11. Alber, J. (2003). Exact algorithms for NP hard problems on networks: design, analysis and implementation (Doctoral dissertation, Tübingen, Univ.).
12. Lanza-Gutierrez, J. M., Gomez-Pulido, J. A., Vega-Rodriguez, M. A., & Sánchez, J. M. (2011). A multi-objective network design for real traffic models of the internet by means of a parallel framework for solving np-hard problems. In 2011 Third World Congress on Nature and Biologically Inspired Computing (pp. 137-142). IEEE.
13. Zhao, H., Chen, Z. G., Zhan, Z. H., Kwong, S. & Zhang, J. Multiple populations co-evolutionary particle swarm optimization for multi-objective cardinality constrained portfolio optimization problem. *Neurocomputing* **430**, 58–70 (2021).
14. Erwin, K. & Engelbrecht, A. Meta-heuristics for portfolio optimization. *Soft. Comput.* **27**(24), 19045–19073 (2023) (**Springer**).
15. Giannakouris, G., Vassiliadis, V., & Dounias, G. (2010). Experimental study on a hybrid nature-inspired algorithm for financial portfolio optimization. In *Artificial Intelligence: Theories, Models and Applications: 6th Hellenic Conference on AI, SETN 2010, Athens, Greece, May 4-7, 2010. Proceedings 6* (pp. 101–111). Springer.
16. Nikoloski, Z., Grimbs, S., May, P. & Selbig, J. Metabolic networks are NP-hard to reconstruct. *J. Theor. Biol.* **254**(4), 807–816 (2008) (**Elsevier**).
17. Nagarajan, N. & Pop, M. Parametric complexity of sequence assembly: theory and applications to next generation sequencing. *J. Comput. Biol.* **16**(7), 897–908 (2009) (**Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA.**).
18. He, D., Choi, A., Pipatsrisawat, K., Darwiche, A. & Eskin, E. Optimal algorithms for haplotype assembly from whole-genome sequence data. *Bioinformatics* **26**(12), i183–i190 (2010) (**Oxford University Press**).
19. Ilango, R., Loff, B., & Carboni Oliveira, I. (2020). NP-hardness of circuit minimization for multi-output functions. In *CCC'20: Proceedings of the 35th Computational Complexity Conference* (pp. 1–36).
20. Ning, W. Strongly NP-hard discrete gate-sizing problems. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **13**(8), 1045–1051 (1994) (**IEEE**).
21. Tkatek, S., Bahti, O., Lmzouari, Y., & Abouchabaka, J. (2020). Artificial intelligence for improving the optimization of NP-hard problems: A review. *Int. J. Adv. Trends Comput. Sci. Appl.*, 9(5).
22. Chinchuluun, A. & Pardalos, P. M. A survey of recent developments in multiobjective optimization. *Ann. Oper. Res.* **154**(1), 29–50 (2007) (**Springer**).
23. Hochba, D. S. Approximation algorithms for NP-hard problems. *ACM SIGACT News* **28**(2), 40–52 (1997) (**ACM New York, NY, USA.**).
24. Kumar, S. N. & Panneerselvam, R. *A survey on the vehicle routing problem and its variants* (Scientific Research Publishing, 2012).

25. D-Wave Systems. (2024, May 31). Retrieved from https://www.dwavesys.com/
26. Bertuzzi, A., Ferrari, D., Manzalini, A., & Amoretti, M. (2024). Evaluation of quantum and hybrid solvers for combinatorial optimization. arXiv:2403.10455.
27. Bożejko, W., Burduk, A., Pempera, J., Uchroński, M., & Wodecki, M. (2024). Optimal solving of a binary knapsack problem on a D-Wave quantum machine and its implementation in production systems. *Ann. Oper. Res.*, 1–16. Springer.
28. Lucas, A. Ising formulations of many NP problems. *Front. Phys.* **2**, 74887 (2014).
29. Pelofske, E., Bärtschi, A. & Eidenbenz, S. Short-depth QAOA circuits and quantum annealing on higher-order ising models. *NPJ Quant. Inf.* **10**(1), 30 (2024).
30. Jaumà, G., García-Ripoll, J. J. & Pino, M. Exploring quantum annealing architectures: A spin glass perspective. *Adv. Quant. Technol.* **7**(4), 2300245 (2024).
31. Nakamura, M., Kaneshima, K. & Yoshida, T. Petri net modeling for Ising model formulation in quantum annealing. *Appl. Sci.* **11**(16), 7574 (2021).
32. Ikeda, H., & Yamazaki, T.: Multi-objective Optimization Technique Based on QUBO and an Ising Machine, *IEEE Access*, (2024), IEEE.
33. Oshiyama, H. & Ohzeki, M. Benchmark of quantum-inspired heuristic solvers for quadratic unconstrained binary optimization. *Sci. Rep.* **12**(1), 2146 (2022) (**Nature Publishing Group UK London**).
34. Brown, Robin, Neira, David E Bernal, Venturelli, Davide, & Pavone, Marco. (2022) Copositive programming for mixed-binary quadratic optimization via Ising solvers, arXiv preprint arXiv:2207.13630.
35. Feld, S. et al. A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. *Front. ICT* **6**, 13 (2019).
36. Dave, K., Nikilesh, B.R., Patel, A., Lalwani, J. Efficient earth observation satellites mission planning with quantum algorithm.
37. Swain, T. Optimisation of active space debris removal missions with multiple targets using quantum annealing. arXiv preprint arXiv:2311.01852, (2023).
38. Makarov, A., Taddei, M.M., Osaba, E., Franceschetto, G., Villar-Rodriguez, E., & Oregi, I. Optimization of Image Acquisition for Earth Observation Satellites via Quantum Computing. In International conference on intelligent data engineering and automated learning, pp. 3–14. Springer (2023).
39. Weinberg, S. J., Sanches, F., Ide, T., Kamiya, K. & Correll, R. Supply chain logistics with quantum and classical annealing algorithms. *Sci. Rep.* **13**(1), 4770 (2023) (**Nature Publishing Group UK London**).
40. Choong, H.Y., Kumar, S., & Van Gool, L. Quantum annealing for single image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1150–1159. (2023).
41. Bucher, D., Nüßlein, J., O'Meara, C., Angelov, I., Wimmer, B., Ghosh, K., Cortiana, G., & Linnhoff-Popien, C. Dynamic Price Incentivization for Carbon Emission Reduction using Quantum Optimization. arXiv preprint arXiv:2309.05502, (2023).
42. Sharabiani, M.T.A., Jakobsen, V.B., Jeppesen, M., & Mahani, A.S. Quantum annealing continuous optimisation in renewable energy. arXiv preprint arXiv:2105.11322, (2021).
43. Fernández-Campoamor, M., O'Meara, C., Cortiana, G., Peric, V., & Bernabé-Moreno, J. Community detection in electrical grids using quantum annealing. arXiv preprint arXiv:2112.08300, (2021).
44. Nembrini, R., Ferrari, D. M. & Cremonesi, P. Feature selection for recommender systems with quantum computing. *Entropy* **23**(8), 970 (2021).
45. Mugel, S. et al. Hybrid quantum investment optimization with minimal holding period. *Sci. Rep.* **11**(1), 19587 (2021).
46. Palmer, S., Sahin, S., Hernandez, R., Mugel, S., & Orus, R. Quantum portfolio optimization with investment bands and target volatility. https://arxiv.org/abs/2106.06735.
47. Mugel, S. et al. Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks. *Phys. Rev. Res.* **4**(1), 013006 (2022).
48. Kim, M., Venturelli, D., & Jamieson, K. Leveraging quantum annealing for large MIMO processing in centralized radio access networks. In Proceedings of the ACM special interest group on data communication, pp. 241–255. (2019).
49. Ding, Y., Chen, X., Lamata, L., Solano, E. & Sanz, M. Implementation of a hybrid classical-quantum annealing algorithm for logistic network design. *SN Comput. Sci.* **2**, 1–9 (2021) (**Springer**).
50. Singh, A., Lin, C.-Y., Huang, C.-I. & Lin, F.-P. Quantum annealing approach for the optimal real-time traffic control using QUBO. In *Proceedings of the 2021 IEEE/ACIS 22nd International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp. 74–81. IEEE (2021).
51. Chitty, D. M., Charles, J., Moraglio, A., & Keedwell, E. Applying a Quantum Annealer to the Traffic Assignment Problem. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pp. 814–822, (2024).
52. Enan, A., Salek, M. S., Chowdhury, M., Comert, G., Khan, S. M., & Majumder, R. Optimal Traffic Flow in Quantum Annealing-Supported Virtual Traffic Lights. *arXiv preprint arXiv:2412.18776*, (2024).
53. Hussain, H., Javaid, M. B., Khan, F. S., Dalal, A. & Khalique, A. Optimal control of traffic signals using quantum annealing. *Quant. Inf. Process.* **19**, 1–18 (2020).
54. Haba, R. et al. Routing and scheduling optimization for urban air mobility fleet management using quantum annealing. *Sci. Rep.* **15**(1), 4326 (2025).
55. Dixit, V. V. & Niu, C. Quantum computing for transport network design problems. *Sci. Rep.* **13**(1), 12267 (2023).
56. Inoue, D. Okada, Akihisa, Matsumori, Tadayoshi, Aihara, Kazuyuki, and Yoshida, Hiroaki. Traffic signal optimization on a square lattice with quantum annealing. *Sci. Rep.* **11**(1), 3303 (2021) (**Nature Publishing Group UK London**).
57. Neukart, F. et al. Traffic flow optimization using a quantum annealer. *Front ICT* **4**, 29 (2017).
58. Fattah, M. A., Morshed, S. R. & Kafy, A.-A. Insights into the socio-economic impacts of traffic congestion in the port and industrial areas of Chittagong city, Bangladesh. *Transp. Eng.* **9**, 100122 (2022).
59. González-Aliste, P., Derpich, I. & López, M. Reducing urban traffic congestion via charging price. *Sustainability* **15**(3), 2086 (2023).
60. Huang, Z. & Loo, B. P. Y. Urban traffic congestion in twelve large metropolitan cities: A thematic analysis of local news contents, 2009–2018. *Int. J. Sustain. Transp.* **17**(6), 592–614 (2023).
61. INRIX Research. 2023 INRIX Global Traffic Scorecard. INRIX, 2022. [Online]. Available: https://inrix.com/scorecard/. [Accessed: 20-Jun-2024].
62. Seong, J., Kim, Y., Goh, H., Kim, H. & Stanescu, A. Measuring traffic congestion with novel metrics: A case study of six US metropolitan areas. *ISPRS Int. J. Geo Inf.* **12**(3), 130 (2023).
63. The Washington Post. Traffic congestion is returning to pre-COVID levels, study says. January 10, 2023. https://www.washingtonpost.com/transportation/2023/01/10/traffic-congestion-inrix-covid/.
64. Qadri, S. S. S. M., Gökçe, M. A. & Öner, E. State-of-art review of traffic signal control methods: Challenges and opportunities". *Eur. Transp. Res. Rev.* **12**, 1–23 (2020).
65. Łach, Ł & Svyetlichnyy, D. Comprehensive review of traffic modeling: towards autonomous vehicles. *Appl. Sci.* https://doi.org/10.3390/app14188456 (2024).
66. Li, J., Chunhui, Y., Shen, Z., Su, Z. & Ma, W. A survey on urban traffic control under mixed traffic environment with connected automated vehicles. *Transp. Res. C Emerg. Technol.* **154**, 104258. https://doi.org/10.1016/j.trc.2023.104258 (2023).
67. Garg, M. & Bouroche, M. Can connected autonomous vehicles improve mixed traffic safety without compromising efficiency in realistic scenarios?. *IEEE Trans. Intell. Transp. Syst.* **24**(6), 6674–6689. https://doi.org/10.1109/TITS.2023.3238889 (2023) (**IEEE**).

68. Yarkoni, S., Neukart, F., Tagle, E.M.G., Magiera, N., Mehta, B., Hire, K., Narkhede, S., Hofmann, M. Quantum shuttle: traffic navigation with quantum computing. In *Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures and Paradigms for Engineering Quantum Software*, 2020, pp. 22–30.
69. Buffoni, L. & Campisi, M. Thermodynamics of a quantum annealer. *Quant. Sci. Technol.* **5**(3), 035013 (2020).
70. Heng, S., Kim, D., Kim, T. & Han, Y. How to solve combinatorial optimization problems using real quantum machines: A recent survey. *IEEE Access* **10**, 120106–120121 (2022) (**IEEE**).
71. Yulianti, L. P. & Surendro, K. Implementation of quantum annealing: A systematic review. *IEEE Access* **10**, 73156–73177 (2022) (**IEEE**).
72. Prakash, K.B. Quantum Meta-Heuristics and Applications. In *Cognitive Engineering for Next Generation Computing: A Practical Analytical Approach* (pp. 265–297). Wiley Online Library. (2021)
73. Salloum, H., Aldaghstany, H. S., Orabi, O., Haidar, A., Bahrami, M. R., & Mazzara, M. (2024). Integration of machine learning with quantum annealing. In *International Conference on Advanced Information Networking and Applications* (pp. 338–348). Springer.
74. Yarkoni, S., Raponi, E., Bäck, T. & Schmitt, S. Quantum annealing for industry applications: Introduction and review. *Rep. Prog. Phys.* **85**(10), 104001 (2022) (**IOP Publishing**).
75. Nath, R. K., Thapliyal, H. & Humble, T. S. A review of machine learning classification using quantum annealing for real-world applications. *SN Comput. Sci.* **2**(5), 365 (2021) (**Springer**).
76. Sun, J. & Lu, S. On the time-independent Hamiltonian in real-time and imaginary-time quantum annealing. *Chin. Phys. B* **29**(10), 100303 (2020) (**IOP Publishing**).
77. Morita, S. & Nishimori, H. Convergence of quantum annealing with real-time Schrödinger dynamics. *J. Phys. Soc. Jpn.* **76**(6), 064002 (2007) (**The Physical Society of Japan**).
78. Morita, S., & Nishimori, H. *Mathematical foundation of quantum annealing. J. Math. Phys*, 49(12), (2008)
79. Dutt, A. & Kar, S. Smooth double barriers in quantum mechanics. *Am. J. Phys.* **78**(12), 1352–1360 (2010) (**AIP Publishing**).
80. Lalus, H. F., Yudhawardana, H. & Aryani, N. P. WKB approximation for analyzing quantum tunneling effect through negative Kratzer potential. *J. Phys. Confer. Ser.* **1918**(2), 022026 (2021).
81. Bertlmann, R. A., & Friis, N. (2008). *Theoretical physics T2: Quantum mechanics*. T2–Script of Sommersemester.
82. Das, A. Quantum annealing and analog quantum computation lecture notes in physics 679. In *Lecture Notes in Physics, 679* (eds Das, A. & Chakrabarti, B. K.) (Springer, Berlin, 2005).
83. McMahon, D. *Quantum computing explained* (Wiley, New York, 2007).
84. D-Wave Systems, Inc. *Quantum Annealing*, D-Wave Documentation, available at: https://docs.dwavesys.com/docs/latest/c_qpu_annealing.html, Accessed 23 Oct 2024.
85. McGeoch, C. & Farré, P. Advantage processor overview. *D-Wave systems* 14–105 (2022).
86. Boothby, K., Bunyk, P., Raymond, J., & Roy, A. Next-generation topology of D-Wave quantum processors, arXiv preprint arXiv:2003.00133.
87. D-Wave Systems Inc., *Topology Introduction: Zephyr*, https://docs.dwavesys.com/docs/latest/c_gs_4.html#topology-intro-zephyr
88. McGeoch, C., Farre, P., Bernoudy, W. *D-Wave hybrid solver service+ Advantage: Technology update*. Tech. Rep., D-Wave Systems (2020).
89. Tambunan, T.D., Suksmono, A.B., Edward, I.J.M., Mulyawan, R. Quantum annealing for vehicle routing problem with weighted segment. In *AIP Conference Proceedings*, volume 2906, number 1, AIP Publishing (2023).
90. Jun, K. QUBO formulations for a system of linear equations. *Results Control Optim.* **14**, 100380 (2024) (**Elsevier**).
91. Boeing, G. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Comput. Environ. Urban Syst.* **65**, 126–139 (2017) (**Elsevier**).
92. Thomas, H., et al. Introduction to Algorithms. (2009).
93. Ngo, T.-G., Dao, T.-K., Thandapani, J., Nguyen, T.-T., Pham, D.-T., Vu, V.-D. Analysis of urban traffic vehicle routing based on dijkstra algorithm optimization. In *Communication and Intelligent Systems: Proceedings of ICCIS 2020*, pp. 69–79, (2021). Springer.
94. D-Wave Systems Inc., *Programming the D-Wave QPU: Parameters for Beginners*, 2023. Available: https://www.dwavequantum.com/media/qvbjrzgg/guide-2.pdf. Accessed April 15, 2025.
95. Natarajan, S. et al. An extension of the Wilcoxon rank sum test for complex sample survey data. *J. R. Stat. Soc. Ser. C Appl. Stat.* **61**(4), 653–664 (2012).
96. Quinton, F. A., Myhr, P., Barani, M., Crespo del Granado, P. & Zhang, H. Quantum annealing applications, challenges and limitations for optimisation problems compared to classical solvers. *Sci. Rep.* **15**(1), 12733. https://doi.org/10.1038/s41598-025-96220-2 (2025).

Acknowledgements

All authors were supported by the Research Center of the Artificial Intelligence Institute at Innopolis University. Financial support was provided by the Ministry of Economic Development of the Russian Federation (No. 25-139-66879-1-0003).

Author contributions

H.S. wrote the original manuscript, while H.S. and A.A. were responsible for editing. The methodology was developed by H.S. and S.Z., with H.S. contributing more to the theoretical aspects and S.Z. focusing on the experimental part. Project supervision was carried out by H.S. and Y.K. All authors reviewed the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to H.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.