Journal of Applied Data Sciences
Vol. 7, No. 1, January 2026, pp. 686-702

ISSN 2723-6471
686

Quantum-Inspired Optimization for Traffic Congestion: A QUBO-Based Approach with Simulated Annealing

Toufan Diansyah Tambunan[1,*], Andriyan Bayu Suksomono[2], Ian Joseph Matheus Edward [3], Rahmat Mulyawan[4]

[1]Doctoral Program of Electrical Engineering and Informatics, School of Electrical Engineering and Informatics, Institut Teknologi Bandung. Jl. Ganesha, 10, Bandung, 40132, Jawa Barat, Indonesia

[1]Multimedia Engineering Technology Study Program, School of Applied Science, Telkom University. Jl. Telekomunikasi no.1 Bandung,40257. Indonesia

[2,3,4]School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jl. Ganesha, 10, Bandung, 40132, Jawa Barat, Indonesia

(Received: July 25, 2025; Revised: September 25, 2025; Accepted: December 22, 2025; Available online: January 31, 2026)

Abstract

Urban traffic congestion remains a persistent challenge, especially when road segments exceed vehicle capacity, leading to increased travel times and road density. This study introduces a new QUBO framework designed to dynamically reduce congestion by optimizing vehicle routes while considering the capacity constraints of road segments. The proposed model establishes quadratic penalties for road segments that exceed the set capacity thresholds, providing incentives to redistribute vehicles to alternative routes while maintaining overall traffic flow efficiency. The QUBO formulation also incorporates road density as a factor to distribute vehicle routing more evenly. The challenge is to ensure that the chosen route does not create potential congestion on the next road segment. We conducted the simulation on a road network consisting of 15 segments (edges) to effectively manage up to 21 vehicles in dense traffic. This QUBO model was created using a quantum annealing approach, but its execution was carried out on an annealing simulation with the Fixstars Amplify and D-Wave Neal machines. The results indicate that the proposed QUBO congestion model can maintain road segment density between 60% and 80% across almost all segment routes. The QUBO congestion model is capable of distributing vehicles evenly, with a Gini coefficient reaching 0.0496 (in an experiment with 21 vehicles), which has the potential to reduce vehicle congestion on road segments. In addition, this model is also capable of avoiding segment choices that exceed road capacity, which is expected to reduce vehicle congestion. Therefore, the resulting QUBO model can be applied to QA engines to reduce congestion on road segments.

Keywords: Traffic Congestion, QUBO, Quantum Annealing, Simulated Annealing, Vehicle Routing

1. Introduction

Traffic congestion has become a critical challenge in urban areas, resulting in economic inefficiencies, increased carbon emissions, and a decline in the quality of life for the community. According to recent research, the QUBO (Quadratic Unconstrained Binary Optimization)-based method provides revolutionary solutions for vehicle routes and traffic signal optimization, reducing wait times and increasing traffic throughput [1], [2]. The QUBO formulation transforms congestion problems into mathematical models by optimizing binary variables, such as traffic light signal durations or route choices, using quadratic cost functions [3]. For example, at the Dongda-Keyuan intersection in Taiwan, Singh [4] demonstrated that average vehicle speed increased when compared to fixed-cycle methods by integrating real-time traffic data into the QUBO model. This advantage stems from QUBO's ability to dynamically adjust signal durations in response to vehicle volume. Additionally, the QUBO model has been shown to reduce waiting times in high-density scenarios, according to simulations conducted using the SUMO (Simulation of Urban Mobility) on a realistic map of Aomori City, Japan [1]. These results demonstrate that QUBO works well in both artificial and complex road conditions, including T-junctions and multi-branch intersections. The QUBO formulation allows various NP-hard

*Corresponding author: Toufan Diansyah Tambunan (tambunan@telkomuniversity.ac.id)
DOI: https://doi.org/10.47738/jads.v7i1.905

ISSN 2723-6471

problems to be mapped into a standard form that can be handled by solution algorithms such as QA (Quantum Annealing). QA becomes a versatile tool for tackling various difficult computational problems, as long as they can be mapped to the structure of the Ising model or equivalently, to the QUBO problem [5]. The problem is mathematically expressed as the minimization of the subsequent objective function [6], [7]:

$$\min_{x\in\{0,1\}^n} x^T Qx \tag{1}$$

In this context, Q represents an $n \times n$ symmetric matrix that contains the interaction coefficients between variables, while x denotes a vector of binary decision variables, where each component of x_i can be either 0 or 1. The goal of QUBO is to determine the binary vector x that minimizes the value of the objective function. There are no explicit constraints in the standard QUBO formulation, as constraints can be integrated into the Q matrix through penalty techniques making it unconstrained. The transformation of real-world problems into QUBO is the most critical and demanding phase. The first step is to determine binary choice variables in the problem, then create the Q matrix (the objective function), and finally resolve the optimization constraints that need to be added to the objective function. The QUBO solution method is carried out using several approaches, including metaheuristic algorithms (such as Simulated Annealing, Tabu Search, and Genetic Algorithm), Quantum Annealing, and Quantum Approximate Optimization Algorithm (QAOA) for gate-based quantum computing [2].

Quantum-inspired techniques like SA (Simulated Annealing), LQA (Local Quantum Annealing) and QA have become effective substitutes for physical quantum hardware in the context of QUBO problem solving. Simulated annealing can be used as a classical benchmark to compare the performance of quantum-inspired algorithms (such as SimCIM) and quantum annealing in solving QUBO-based optimization problems. While QA uses the quantum tunneling principle to avoid local minima, SA uses thermodynamic process simulation to explore the solution space [8]. Exploring low-energy configurations of quantum mechanics through superposition and tunneling may be advantageous due to the optimization required to prevent local minima traps. The quantum annealing mechanism operates on this principle and has been proposed as an algorithm capable of addressing optimization challenges, including the determination of the ground state of a spin Hamiltonian [9]. Quantum computers, exemplified by D-Wave, have demonstrated efficacy in addressing numerous NP-hard issues. A quantum annealer operates in a manner similar to the resolution of optimization issues by classical simulated annealing. A multivariate function formulates the energy landscape, ensuring that the ground state aligns with the problem's solution [10].

According to a comparative study, the SimCIM algorithm, a GPU-based QA application, improves accuracy on 100-node graphs and generates more optimal solutions than traditional solvers like Gurobi [11]. The requirement for minor embedding and the inability to obtain optimal solutions in large-scale test cases continue to pose obstacles to implementing QA on D-Wave hardware. However, despite the need for parameter adjustments like annealing time and temperature range, the combination of SA and the gradient descent mechanism (GD-QIA) consistently yields near-optimal solutions for the VRP (Vehicle Routing Problem) [6], [12]. Although hybrid methods on D-Wave improve performance by using classical solutions as starting points, they are still not as fast as pure SA [13]. These results highlight how quantum-inspired algorithms can serve as a bridge to fully quantum applications while remaining compatible with traditional computing infrastructure.

Neukart et al. initiated the use of quantum annealing for traffic flow optimization, illustrating the mapping of real-world traffic issues to QUBO formulations compatible with QPUs (Quantum Processing Units) [14]. Their hybrid quantum-classical approach showed promising results in redistributing traffic to reduce congested roads. Tambunan et al.[15] refined this development by adding weighted road segments to the QUBO model. This change allows optimization to take into account various road features, such as distance, priority, and capacity. Their research shows that segment weighting has a significant impact on how vehicles choose routes, facilitating traffic route allocation based on road conditions. Salloum et al. built this foundation to create the Q-CFTO (Quantum Congestion-Focused Traffic Optimization) method. This method cleverly breaks down complex traffic problems into smaller subproblems that can be directly embedded into the QPU [16]. Their method makes a significant difference in how quickly a computer can perform tasks. For example, this method reduces processing time and shows that traffic congestion can be reduced in a simulation of 500 vehicles.

This paper presents a QUBO formulation to manage vehicle traffic flow to identify the correct route and avoid overlap with other vehicles. We will provide alternative route solutions to avoid congestion, limiting this problem to a number of vehicles with the same destination. Our traffic flow model takes into account the weight of road segments and density capacity limits when determining vehicle routes. The cost value will affect the density level of the selected road segment to avoid traffic congestion. Meanwhile, the constraint will be defined as the condition of vehicles sharing a lane at a specific time. We obtain the cost function from the number of vehicles sharing a single lane. Thus, we can achieve the goal of minimizing the number of overlapping segments on each vehicle's route. We tackle this challenge by defining it using the QUBO model, a Quantum-inspired approach. Quantum annealing presents a promising approach for addressing complex combinatorial optimization problems, such as vehicle routing, by determining the physical ground state of a quantum system. To utilize this paradigm, real-world problems must initially be converted into an appropriate mathematical framework. The QUBO model serves as this crucial bridge, providing a standard formulation for quantum annealers. This formulation is the source of the quantum aspect in our work. While a physical quantum annealer can solve QUBOs, access remains limited. Fortunately, powerful classical algorithms exist to find high-quality solutions to these QUBO problems on conventional hardware. SA stands out as a prominent and effective classical heuristic for this task. Therefore, our methodology constitutes a quantum-inspired approach: we use a quantum formulation (with QUBO) to structure the problem and then employ a sophisticated classical solver (SA) to find the optimal solution. The subsequent phase will entail the optimization process through annealing utilizing Simulated Annealing techniques, such as D-Wave Neal and Fixstars Amplify.

The discussion in this paper will be divided into several sections. Section 2 will explain the research methods employed and the development of the QUBO model to address traffic congestion. Experiments on the annealer machine and their results will be presented in section 3, and the analysis of the traffic congestion optimization results will also be discussed. Then, section 4 presents a conclusion that interprets the research results and discusses opportunities for further research development.

2. Methods

The research process employed several methods and procedures to address traffic congestion issues. Among them is the exploration of the QUBO formula in previous research, which has the potential to cause road congestion issues. Then, conduct a small-scale vehicle network simulation with a dynamic graph, and evaluate the model using the D-Wave Neal (SA) and Fixstars Amplify (AE) annealer machines.

2.1. Quantum Annealing for Traffic Optimization

Quantum annealing is a quantum computing paradigm specifically designed to solve optimization problems. Unlike SA, which relies on thermal fluctuations, QA utilizes quantum mechanical phenomena such as superposition and quantum tunneling [17]. The system starts in a superposition of all possible states with a simple initial Hamiltonian H_D a transverse field Hamiltonian that causes the superposition [18], [19].

$$H_D = -\sum_{i=1}^{N} \Gamma_i \sigma_i^x \tag{2}$$

The system then adiabatically transforms the initial Hamiltonian into the problem Hamiltonian H_P, which problem represents the QUBO/Ising model.

$$H_P = \sum_{i<j}^{N} J_{ij} \sigma_i^z \sigma_j^z + \sum_{i=1}^{N} h_i \sigma_i^z \tag{3}$$

$\sigma_{i,j}^{x,z}$ are Pauli matrices operating on a qubit (N), h_i is the qubit biases (magnetic strengths) and J_{ij} is coupling strengths (interaction strength between spin). By maintaining a sufficiently slow adiabatic process, the quantum system tends to remain in the ground state (lowest energy) of the evolving Hamiltonian, which will ultimately become the optimal solution for the QUBO problem. The coupling strength (J_{ij}) between qubits in H_P is a form of the Q_{ij} coefficient in QUBO, while the local bias (h_i) on each qubit in H_P can be interpreted as the Q_{ii} coefficient in QUBO. Minimizing the QUBO function is equivalent to finding the spin configuration that yields the lowest energy value of the

Hamiltonian H_P. The matrix QUBO is defined in the upper-triangular form by the optimization function [5] that is presented below, where $x_{i,j} \in \{0,1\}$:

$$f(x_1,\ldots,x_n) = \sum_{i=1}^{n} Q_{ii}\, x_i \;+ \sum_{1\le i<j\le n} Q_{ij}\, x_i x_j \tag{4}$$

Afterwards, from the resulting QUBO matrix, the binary variable values will be sought that, when combined, can yield the most minimal objective function equation result. This process involves utilizing optimization algorithms to explore the solution space efficiently. Recent research has extensively examined the application of QUBO models on quantum annealing machines for addressing optimization challenges [2]. The resolution of traffic issues is a prominent topic in quantum annealing research, as shown by the use of QA machines like D-Wave or quantum-inspired SA solvers such as Neal, Fujitsu Digital Annealer, Gurobi, or Fixstars Amplify AE [20]. Quantum annealing or simulated annealing can address problems formulated as QUBO or Ising models. The identification of QUBO model for the Travelling Salesman Problem (TSP) and the VRP are two of the most dominant research areas in the application of quantum computing to routing problems [21]. These problems have attracted significant attention due to their inherent computational complexity and their broad applicability in various real-world domains, ranging from logistics to smart cities. Prior research concerning traffic flow optimization [14] has an objective function focused on minimizing the quantity of cars selecting identical routes. Consequently, it enhances overall system performance by utilizing road density costs and capacity limitations. The research [15] was subsequently developed by incorporating road segment weights, which can be interpreted as distance, travel time, or other route prioritization factors. The following delineates the cost function for determining the vehicle's chosen route on the road segment:

$$\text{cost}(s_m) = \left(\sum_{(i,j)\,\in\, R_{s_m}} w_{ij}\, q_{ij}\right)^2 \tag{5}$$

The cost function of each segment $cost(s_m)$, is calculated from the number of vehicles ($i = \{1,2,\ldots,n\}$) using that segment. The variable q_{ij} will represent a qubit and will represent the combination of each route choice (j) of the vehicle (i). By accumulating the weights w_{ij} on each segment that is the preferred route for vehicle q_{ij}, the configuration of route choices (R_s) can be used to construct an overall cost function equation. Minimizing the number of segments that overlap with other vehicles is essential for optimizing vehicle traffic. The density level of the designated road segment will be influenced by the cost value in order to alleviate congestion.

$$constraint = \sum_{i=1}^{n}\left(\sum_{j=1}^{3} q_{ij} - 1\right)^2 \tag{1}$$

The QUBO form is completed by the constraint function, which is valued at 0, indicating that the condition is true when only one vehicle route option is active ($q_{ij} = 1$) [14]. The vehicle traffic flow problem is defined by the requirement that each vehicle (i) must select precisely one route option (j). The two equations mentioned above (5) (6), will generate a QUBO matrix, representing the vehicle routing problem. This matrix will subsequently be subjected to an annealing process to determine the most optimal route. In the visualization below (figure 1), the matrix form QUBO consists of linear (diagonal) and quadratic (upper) terms, which generates routes for n=21 vehicles.

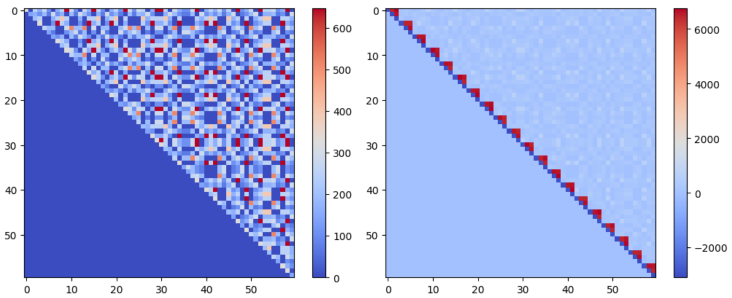

Figure 1. Visualization q_{ij} For N=21 Vehicles (a), And QUBO Matrix (a) For Vehicle Routing

The QUBO matrix developed for the identification of traffic optimization problems can subsequently be utilized with a quantum annealer, such as the D-Wave QPU 2000Q, or it can be executed on a SA machine using the same QUBO model. Optimal results (q_{ij}) are achieved when the route configuration for all vehicles is valid, minimizing total traffic costs while ensuring the lowest energy consumption. Another traffic modeling strategy emphasizes the direct correlation between the volume of vehicles traveling on a roadway and traffic congestion [16]. The redistribution of vehicle traffic within the zone under consideration will be influenced by the congestion weight at intersections that are not within the optimization area. Nevertheless, the concept of determining road congestion costs is generally shared between cost modeling and traffic optimization constraints.

3.1.Congestion Problem

The problem of vehicle routing in previous studies [14], [15], was defined as a combination of route choices for each vehicle q_{ij}, which crosses a road segment from the starting point to the ending point. Our congestion optimization of traffic flows is carried out by minimizing the number of vehicles traversing a road segment simultaneously. Each vehicle is given a scenario of choosing one of three route choices ($j = \{1,2,3\}$), which would remain mathematically identical if given more alternative route choices. As shown in equations (5) and (6), giving each vehicle three route choices are done to simplify the mathematical description and to focus on demonstrating the feasibility of the congestion model. Therefore, in this discussion, a road network model in the form of a graph (node, edge) is created that is interconnected by giving weight (w) to the edge. The road network dataset is mapped to a graph (nodes, edges), and vehicle trajectories are randomly generated, with some vehicle routes having different start and end points. This procedure further increases complexity due to the possibility of vehicles overlapping in different directions. This study uses a road network with 10 nodes as intersection points and 15 edges as segments. When implementing QUBO in optimization in an annealing machine, the road network representation is mapped to a graph (containing nodes and edges). Similar to urban road conditions, this graph representation contains branching roads, and route segments may overlap between vehicles. The primary purpose of using this controlled and representative network is to focus our investigation on the feasibility of the proposed QUBO model for managing traffic congestion. The density of each road segment is displayed in figure 2, provided that the route that was taken was the first-choice route ($j = 1$) from the inventory of vehicle trajectory data collection (variable T on the edge). This graph shows the difference in road density when there are more vehicles ($n = 21$), and they only accumulate in certain segments.

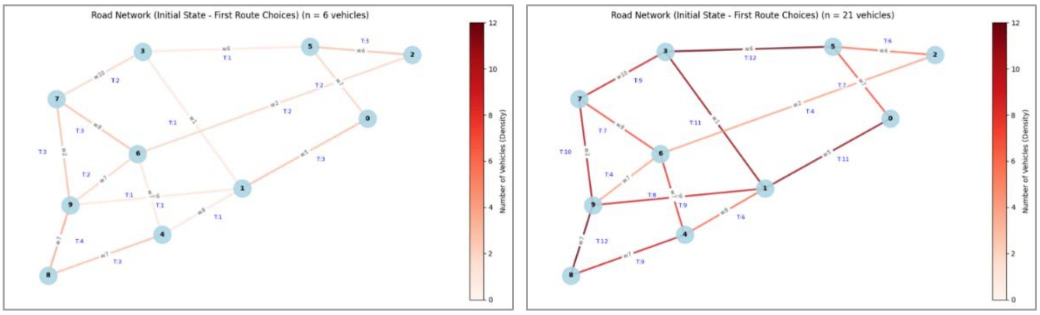

Figure 2. Visualizing The Random Road Network Graph (node, edge), Segment Weights, Vehicle Trajectories And Density

The road density dataset contains vehicle trajectory data from the number of vehicles n = 5 to n = 21. The graph used as a road network has the same number of segments, namely 10 nodes with 15 edges. Each edge represents a connection between two nodes, allowing for the simulation of various traffic scenarios (see example in table 1). The network traffic flow patterns and congestion dynamics can be better understood by comparing vehicle trajectories at various densities. The congestion problem in this data is interpreted as the high density of vehicles on a particular road segment, which results in a greater route weight value. For example, for vehicle $i = 2$, preferred route $j = 3$ has the largest total weight of 49. The result means that the preferred route should be avoided because it will not provide optimal cost.

Table 1. Sample ($n = 5$) Dataset Random Road Network Graph With Vehicle Trajectories

Vehicle	options	node		Route selection	weight	total traffic (T)
(i)	(j)	start	finish		(w_{ij})	
1	1	6	7	6 - 9 - 1 - 0 - 4 - 3 - 8 - 5 - 7	47	20
	2			6 - 2 - 0 - 4 - 1 - 5 – 7	39	12

 ISSN 2723-6471

Vehicle	options	node		Route selection	weight	total traffic (T)
(i)	(j)	start	finish		(w_{ij})	
	3			6 - 2 - 1 - 5 - 3 – 7	33	5
2	1	6	1	6 - 9 - 3 - 8 - 5 – 1	39	12
	2			6 - 9 - 3 - 4 - 0 – 1	38	11
	3			6 - 7 - 5 - 3 - 4 - 0 - 2 - 9 - 1	49	14
3	1	9	0	9 - 2 - 6 - 8 - 5 - 7 - 3 - 4 - 0	47	19
	2			9 - 2 - 1 - 5 - 3 - 4 – 0	41	9
	3			9 - 3 - 8 - 6 - 5 - 1 - 2 – 0	44	9
4	1	3	0	3 - 4 - 1 - 5 - 7 - 6 - 9 - 2 - 0	52	17
	2			3 - 4 - 1 – 0	13	6
	3			3 - 7 - 5 - 6 - 2 - 1 – 0	32	8
5	1	1	5	1 - 0 - 2 - 6 - 8 – 5	31	12
	2			1 - 0 - 4 - 3 - 9 - 2 - 6 – 5	49	12
	3			1 - 4 - 0 - 2 - 9 - 6 - 8 – 5	45	16

Vehicle route determination is carried out systematically in Algorithm 1. This process generates a set of initial route options for each vehicle (i), according to the desired number of alternative routes ($j = 3$). For each vehicle, the algorithm randomly selects a different start node and end node. This algorithm will then find all non-cyclical paths between pairs of nodes connected by segments (edges), with this search process iterating MAX_ATTEMPTS. To ensure route diversity, these paths are randomly shuffled, and as many route options ($R_{options}$) are collected as needed by each vehicle. The first route from this initialization result will be taken as the INIT route, which will later be compared with the results of routing optimization using the QUBO model.

Algorithm 1: Generate Initialization Vehicle Routes

```
INPUT:
  G: Graph (N, E) representing the road network
  V_total: The total number of vehicles
  R_options: The number of alternative route options per vehicle
OUTPUT:
  D_vehicles: A collection of vehicle data with route options

FUNCTION GenerateInitialRoutes(G, V_total, R_options)
  D_vehicles ← an empty collection
  Nodes ← Get all nodes from G
  FOR v FROM 1 TO V_total DO
    vehicle_routes ← an empty list
    unique_paths_found ← an empty list
    attempt_count ← 0
    pair_found ← FALSE
    WHILE attempt_count < MAX_ATTEMPTS AND NOT pair_found DO
      n_start ← Randomly select a node from Nodes
      n_end ← Randomly select a node from Nodes
      IF n_start IS NOT n_end THEN
        all_paths ← FindAllSimplePaths (G, n_start, n_end)
        IF all_paths is not empty THEN
          Randomly shuffle the order of all_paths
          FOR EACH path IN all_paths DO
            IF path is not in unique_paths_found THEN
              Add path to unique_paths_found
            END IF
            IF size of unique_paths_found ≥ R_options THEN
              BREAK
            END IF
```

```
                END FOR
                pair_found ← TRUE
            END IF
        END IF
        Increment attempt_count
    END WHILE
    vehicle_routes ← first R_options elements of unique_paths_found
    WHILE size of vehicle_routes < R_options DO
        Add an empty route to vehicle_routes
    END WHILE
    Store {id: v, start: n_start, end: n_end, routes: vehicle_routes} in D_vehicles
  END FOR
  RETURN D_vehicles
END FUNCTION
```

3.2. QUBO Formula for Traffic Congestion

The total cost of vehicles passing through all segments is calculated to perform QUBO mapping for vehicle traffic problems. The QUBO model is proposed in this study as a solution to the potential congestion caused by the density of vehicles on specific road segments. This is achieved by implementing several objective functions. The cost function (Q_A) minimizes the number of vehicles across all road segments, which are influenced by segment weights that represent distance. Another function (Q_B) is used to avoid selecting routes with high vehicle density levels. The model also includes constraints, where the constraint (Q_{C1}) requires that each vehicle has exactly one selected route solution, and the constraint (Q_{C2}) is to limit the number of vehicles in each segment so it does not exceed the specified capacity. The overall objective function is as follows,

$$QC = Q_A + Q_B + Q_{C1} + Q_{C2} \tag{7}$$

This mapping allows optimization of traffic flow, allowing for more efficient routes and reducing congestion. By analyzing the costs associated with vehicle route choices and segment weights, the solution can improve the overall efficiency of segment routes. The road segment weight affects this cost in equation (5), which can be interpreted as distance, travel time, or other vehicle segment parameters. Cost function (Q_A) form for all segment (S) vehicle routes,

$$Q_A = \alpha \sum_{s \in S} \left(\sum_{(i,j) \in R_s} w_{ij}\, q_{ij} \right)^2 \tag{8}$$

Where q_{ij} is a binary variable $\{0,1\}$ of the vehicle i that choose route j. If the vehicle i chooses route j then $q_{ij} = 1$, and vice versa $q_{ij} = 0$ if the vehicle i does not choose route j. To address traffic congestion issues, the densely populated road segment is subjected to a substantial burden. The cost function of the route is either more optimal or less optimal as the value of the selected road density increases. This effect is the result of the accumulation of vehicles on road segments that are already densely populated (d_s) by vehicles. Consequently, it is imperative to incorporate a function (Q_B) that imposes a penalty if the distribution of vehicles only accumulates on specific segments.

$$Q_B = \beta \sum_{s \in S} \left(\sum_{(i,j) \in R_s} d_s q_{ij} \right)^2 \tag{9}$$

Where d_s is the road density volume on segment S. Vehicle route solutions (q_{ij}) will not be selected if they can cause congestion in other places that have high d_s values. Therefore, this model will distribute traffic density evenly and keep its value low in each segment R_s. The density value d_s can be determined dynamically using traffic condition prediction data related to the vehicle route. Data can be acquired in real-time from sources such as sensors, surveillance cameras, GPS, or other traffic systems. Furthermore, the d_s value can be determined using historical traffic data in the form of average density data, segment density classification (road type or day/night conditions), or a constant density value for each road segment. In addition, the above equation (8) and (9) also assumes the value of the α and β, penalty

ISSN 2723-6471

coefficients, aiming to provide the same weighting factor for congestion in any segment. For example, when the travel time criterion must be a priority, then the α parameter can be tuned larger than β. And vice versa, if the main criterion is an even distribution of road density to reduce congestion, then the β parameter is made larger than α. The next step is to set a condition that each vehicle can only choose one of the three possible routes given. As a result, each vehicle has a combination of route options that have a total value of 1, indicating that only one binary variable q_{ij} has been chosen. We adopt this constraint from the previously used model [14], [15], [22], to precisely select only one optimal solution. The following is a constraint function to meet this condition.

$$Q_{C1} = K \sum_{i=1}^{n} \left(\sum_{j=1}^{3} q_{ij} - 1 \right)^2 \tag{10}$$

Where K is a positive penalty coefficient, which is very large for violation of this constraint, this parameter will ensure the constraint of assigning one route to each vehicle. To ensure that vehicle density is not only focused on one road segment, it is necessary to limit the capacity of each road segment. This capacity (C_s) can be generated randomly, which represents the upper limit of the number of vehicles allowed to use segment s at a time. This limitation is an intrinsic characteristic of the road segment. The goal is to ensure that the resulting route solution is not only optimal in terms of cost (duration) but also feasible in terms of network capacity.

$$Q_{C2} = K_c \sum_{s \in S} \left(\left(\sum_{(i,j) \in R_s} q_{ij} \right) - C_s \right)^2 \tag{11}$$

Not all road segments may have explicit capacity constraints in the model, so the R_s segment allows a focus on crucial segments (e.g., bottlenecks, major intersections, or toll roads). This scenario would lead to traffic patterns that are impractical and do not accurately reflect real-world conditions. By taking into account the capacity constraint, we guarantee that the solution not only optimizes routing but also adheres to the tangible limitations of the infrastructure, resulting in a more practicable and effective traffic management strategy. In the Q_B formula, the equalization target tries to even out the load below capacity, while the capacity constraint Q_{C2} ensures that the absolute limit is never exceeded. The two work together: equalization prevents overconcentration, and capacity prevents violation of the absolute limit. Tuning the parameters β, K, and K_c is crucial here to avoid exceeding the vehicle capacity restrictions per segment in the constraint. A gradual approach is taken in the process of setting the K_c parameter to prevent the number of cars that travel through the route segment from being penalized to an excessive degree. Therefore, when vehicles switch to another route option, it is ensured that the option does not cause congestion elsewhere.

The following are the steps to implement the QUBO model for vehicle route optimization problems [2], [14], [15]. First, the problem is identified by mapping the road network as a graph consisting of nodes and edges, and this includes segment weight data and segment capacity. Next is the route selection for each vehicle that has a starting point and a destination. The vehicle will have a choice of three routes to optimize, and each route choice will globally affect the road density data conditions. This route selection is a combination of binary values on the variable q_{ij} (1 is actively selected, and 0 is not selected). This is followed by the construction of the QUBO matrix, which involves defining the variable q_{ij} with respect to vehicle route choices. The QUBO matrix is also equipped with penalty coefficient values that produce its quadratic and linear form. Afterward, optimization is performed on an Annealer machine with the QUBO matrix as input; this annealing process can use either a QA or SA machine. The final step is mapping the optimization results from the annealer by selecting the lowest energy values and valid routes. The steps above mean figuring out the best route for the vehicle based on optimization criteria by looking at the results of the annealing process. This process makes sure that the chosen route either costs the least or works the best by focusing on the lowest energy value.

3.3. Annealing Engine for QUBO

Quantum annealing research frequently employs quantum processors, including D-Wave. Nevertheless, the QUBO model that has been developed must be validated on other annealing machines to ensure that it is functional, as limited access to quantum computers is also a concern. In addition to the QPU, D-Wave also has Neal [18], which is a simulated annealing optimization mechanism that operates on the CPU. D-Wave utilizes quantum physics principles directly on its hardware (quantum hardware) to perform quantum annealing [23]. This technology enables the simultaneous exploration of numerous states and the potential for quantum tunneling through energy barriers. In contrast, Neal (SA) is a classical algorithm that operates on conventional computers, employing probabilistic simulations to replicate the annealing process. The efficacy of Neal (SA) as a benchmark and comparison tool against quantum annealers such as D-Wave is essential [18]. Neal can simulate the annealing process on classical computing, providing a foundation for performance comparison in terms of solution quality and execution time, while D-Wave QPU utilizes quantum tunneling to expedite the solution search.

Furthermore, to D-Wave Neal (SA), this research will also use Fixstars Amplify AE (annealing engine), which can be an alternative computation in the QUBO model (see comparison in table 2). Fixstars Amplify AE provides a platform and Software Development Kit (SDK) that facilitates the use of annealing machines to solve combinatorial optimization problems. The Amplify AE machine is GPU-based and designed to solve large-scale combinatorial optimization problems (over 100,000 bits) at high speed [20]. As part of its optimization procedure, this machine will use stochastic search to investigate various configurations (including internal samples) throughout the annealing phase. Amplify will return the singular configuration with the lowest energy from all the configurations investigated in a single annealing. Table 2 serves as an explanation of the distinctions between Amplify and Neal, the two annealers. These disparities also influence the technical dimensions of their application in QUBO optimization.

Table 2. Comparison Annealer Tools For QUBO Model, [18], [20]

Features	**Annealing Engine**	
	D-Wave Neal	**Fixstars Amplify AE**
1. Computing	CPU (Software-based)	GPU (Hardware & Software based)
2. Typical Capacity	Limited by local CPU/memory	High-capacity, scalable (cloud GPU)
3. Problem Input	QUBO/Ising models (matrices)	QUBO models built via Amplify SDK
4. Speed	Slower for large problems (CPU-dependent)	Fast for large problems (GPU-accelerated)
5. Max Var/Qubits	Practically limited by system memory/performance	>100,000 (fully connected coupling); millions (sparse)
6. Annealing Type	SA	SA on GPU
7. Accessibility	Open-source, local execution	Commercial (SDK + paid solver access)
8. Coefficient Precision	Standard float precision	Up to 64-bit real numbers
9. Access/Limits	Free access, open-source (local CPU limits)	Free trial (with usage limits); token access

Amplify includes a timeout parameter (measured in seconds) that restricts the processing duration on the Amplify server. In Neal, the num_reads parameter specifies the quantity of sample iterations to be performed on the Neal machine. This experiment will implement a time limit of 5 seconds (timeout=5000) for Amplify. We will use a num_reads=1000 for Neal to extract the optimal solution with the least amount of energy from the outcomes. The results obtained from both Amplify and Neal will be compared to evaluate their effectiveness in solving the QUBO optimization problem. By analyzing the performance, we aim to determine which method yields the most efficient and accurate solutions.

3. Results and Discussion

The computational process in this study begins with the preparation of vehicle flow dataset randomly. We will generate the first dataset in the form of a graph, where nodes represent route points and edges represent route segments (S). Each segment will be assigned a weight (w) that will later be minimized. The second data in the preparation stage is the vehicle routes that will be generated randomly according to the number of vehicles (each vehicle is given 3 route options). We create these routes with different starting and ending points, resulting in diverse route paths (see sample in table 1). Thus, this will make the segment density level increasingly complex. For example, for $n = 5$ vehicles, with $i = \{1,2,3,4,5\}$, and each vehicle having 3 route options $j = \{1,2,3\}$, there will be a total of 15 q_{ij} variables in the QUBO matrix. The next step is to construct the QUBO matrix, which is a representation of the road congestion problem. This step involves mapping the vehicle route data into the QUBO variable q_{ij}. Then the accumulation of each vehicle route will cause a buildup of the number of vehicles on each road segment (S). Equation (7) then supplements this model by incorporating segment weight (w_{ij}) conditions and segment density (d_s). There is also a capacity limit for the number of vehicles in each segment, which is calculated according to the number of vehicles ($C_s = \frac{n}{2}$). This capacity limit applies uniformly to all segments to ensure that vehicles are not concentrated on just two or three segments. Determining a large capacity limit will prevent capacity violations and actively participate in load balancing across the road. This substantial reduction is particularly significant in situations where 15 road segments are occupied by a combination of routes of up to 21 vehicles, which would inevitably lead to vehicle queues.

The mapping of the QUBO matrix can be seen in the example in figure 1, where the visualization of the valid upper-triangular matrix will form the diagonal part as the linear coefficients (q_{ii}) and the upper non-diagonal part as the quadratic coefficients (q_{ij}). These two parameters represent the QUBO model in equation (4), which will later be input into the annealer machine for optimization. The annealer machine will process this QUBO representation to determine the optimal solution by minimizing the energy function defined by the matrix.

$$QW = Q_A + Q_{C1} \tag{12}$$

The first annealing process will be conducted using the QUBO equation (12), that has been used in previous research [14], [15], which will later be used as a comparison. In addition, the initialization of vehicle route choices is done randomly (init) to serve as a comparison against routes generated at random without optimization. We will compare the QUBO model with the weighted segment (QW, equation.12) for the traffic congestion (QC, equation.7) problem. The outcomes of both models QUBO will be analyzed to determine their effectiveness in minimizing traffic congestion. The annealing results on the QUBO model obtained routes for all vehicles. Figure 3 displays the distribution of vehicle density across all available road segments, illustrating the results of the vehicle route selection. In the results of the Init route, it can be observed that the overall results show a vehicle density exceeding the capacity limit (S) of the segment. On the other hand, there are also Init route data where the distribution is uneven, marked by routes that are not filled with vehicles. The QW results yield different routes, each aiming to find the most optimal weight cost (w). Thus, vehicle routes can be directed to road segments with lower weights. This approach is indeed in line with the objective function of the QUBO model QW, which aims to find the minimal weight (w). However, the consequence of such a route model is that segments of the road with low weights will potentially lead to extreme vehicle congestion. This effect can be observed in the QW results for $n = 8$, $n = 12$, $n = 15$, $n = 18$, $n = 19$, and $n = 21$. In terms of the total calculation of the density of all segments and the cost of the weights, it may become minimal, but because vehicles only pile up on certain segments, it will result in congestion. The results of the QUBO congestion model, both QC+Amplify and QC+Neal, show the same outcome. The QC results can effectively manage vehicle distribution on route segments, even relatively evenly under conditions $n = 11$ and $n = 17$. Model QC (Amplify and Neal) successfully distributed vehicles very stably and consistently. In the n=11 route data, it can be seen that more than 50% of the road segments have the exact same number of vehicles (4 vehicles).

Furthermore, the QC model generates routes that prevent vehicles from surpassing the segment capacity. This means that the route selection for vehicles does not create the potential for excessive vehicle congestion. The QUBO congestion (QC) model is designed so that, while it naturally seeks routes with low weight, our penalty terms related to segment density (d_s) and capacity (C_s) act as a balancing mechanism in this problem. When too many vehicles choose the same path with the minimum weight (w_{ij}), the segment density will be high and will incur a significant penalty. This is the case in the QW model, which focuses on optimizing the weight cost of vehicle routes. In contrast, the QC model forces optimization to identify a more globally balanced solution, even if that means utilizing some individually longer route segments. Table 3 shows that for n=18 vehicles, the total weight cost of the QW model is 411, the lowest value compared to other routing methods (Init or QC).

Table 3. Comparative Results Between Vehicle Routing Method

n	Parameters	Result methods			
		Init	QW	QC+Amplify	QC+Neal
5	Gini coefficient	0.2827	0.2603	0.2222	0.2222
	Total weight	154	120	125	125
	AVG density	1.67	1.4	1.4	1.4
	Over capacity	3	1	0	0

ISSN 2723-6471

6	Gini coefficient	0.2632	0.4000	0.1818	0.1818
	Total weight	250	168	188	188
	AVG density	2.53	2.07	2.2	2.2
	Over capacity	5	5	0	0
7	Gini coefficient	0.1848	0.2424	0.1255	0.1255
	Total weight	259	190	206	206
	AVG density	2.2	2	2.27	2.27
	Over capacity	4	2	0	0
8	Gini coefficient	0.3400	0.3179	0.1812	0.1812
	Total weight	246	208	228	228
	AVG density	2.67	2.6	2.6	2.6
	Over capacity	3	2	0	0
9	Gini coefficient	0.2516	0.2427	0.1228	0.1228
	Total weight	344	219	222	222
	AVG density	3.53	2.6	2.53	2.53
	Over capacity	5	1	0	0
10	Gini coefficient	0.2208	0.2411	0.2128	0.2128
	Total weight	370	255	265	265
	AVG density	4.07	3.13	3.13	3.13
	Over capacity	3	1	0	0
11	Gini coefficient	0.1843	0.2914	0.0972	0.0972
	Total weight	413	296	351	351
	AVG density	4.53	3.6	3.93	3.93
	Over capacity	6	3	0	0
12	Gini coefficient	0.1941	0.3359	0.1753	0.1753
	Total weight	395	256	314	314
	AVG density	4.53	3.47	3.6	3.6
	Over capacity	1	1	0	0
13	Gini coefficient	0.2387	0.2573	0.1200	0.1200
	Total weight	382	315	336	336
	AVG density	4.13	3.8	4	4
	Over capacity	3	2	0	0
14	Gini coefficient	0.2074	0.2457	0.1370	0.1370
	Total weight	547	397	428	428
	AVG density	6	4.67	4.87	4.87
	Over capacity	4	2	0	0
15	Gini coefficient	0.1488	0.3747	0.0782	0.0782
	Total weight	525	356	451	451
	AVG density	5.73	3.87	5	5
	Over capacity	2	3	0	0
16	Gini coefficient	0.1407	0.2500	0.2146	0.2146
	Total weight	538	371	383	383
	AVG density	6.07	4.27	4.26	4.27
	Over capacity	0	2	0	0
17	Gini coefficient	0.1634	0.2719	0.1075	0.1075
	Total weight	654	418	554	554
	AVG density	7.4	5.07	6.2	6.2
	Over capacity	4	2	0	0
18	Gini coefficient	0.1545	0.3009	0.0807	0.0807
	Total weight	639	411	488	488
	AVG density	6.73	5.2	5.4	5.4
	Over capacity	1	2	0	0
19	Gini coefficient	0.2070	0.3407	0.0809	0.0809
	Total weight	617	422	549	549
	AVG density	7	5.4	6.27	6.27
	Over capacity	3	3	0	0
20	Gini coefficient	0.1935	0.2611	0.1031	0.1031
	Total weight	635	518	561	561
	AVG density	7.53	6.33	6.47	6.47
	Over capacity	2	3	0	0

21	Gini coefficient	0.1739	0.2370	0.0496	0.0496
	Total weight	730	538	661	661
	AVG density	8.33	6.6	7.53	7.53
	Over capacity	4	1	0	0

However, the consequence is a buildup of vehicles on several road segments. This phenomenon is evident from the Gini coefficient of 0.3009, significantly higher than the QC value of 0.0807. Furthermore, two segments in the QW model experience overcapacity, while this phenomenon does not occur in the QC model. In figure 3, we can observe that for $n = 18$, the number of vehicles in the overcapacity segment reaches 12 vehicles, compared to the 9-vehicle limit capacity (red line).

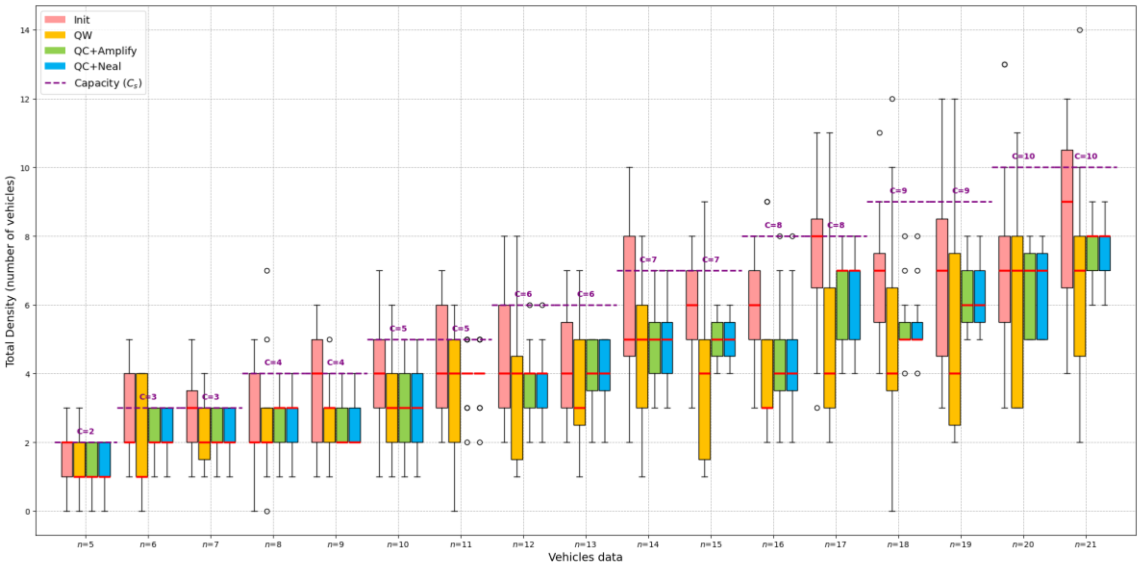

Figure 3. Visualization of Vehicle Density on All Route Segments, Before (init) and After (QW, QC) Optimization

To examine the effect of the penalty coefficients on the optimization results, we conducted a sensitivity analysis on each parameter. We analyzed the effect of each coefficient (α, β, K, K_c) by varying its values significantly and observing their impact on the optimization results. Based on the QUBO equation we proposed, at least an important analysis can be seen in the route weight cost, segment density, exactly one route solution for each vehicle, and segment capacity violations. The results in table 4, show that changes in the penalty coefficient value will affect the optimization results, both for the QW and QC models. These results illustrate the need for caution in setting the value of this parameter to ensure the resulting solution meets its needs. The analysis shows that the β penalty coefficient has the most significant influence on the Gini coefficient, indicating the role of this penalty as a load balancer. Conversely, the α penalty coefficient has the most direct impact on changes in total weight. This analysis confirms that the basic parameters we chose provide a reasonable balance between objectives.

Table 4. Parameter Sensitivity Analysis, for data sample $n = 15$ vehicles

Penalty Coeff.	Parameter	Init	QW	QC+Amplify	QC+Neal
($\boldsymbol{\alpha}$; $\boldsymbol{\beta}$; K; Kc) = (5.5; 6.0; 150; 40)	Gini coefficient	0.161905	0.262687	0.077108	0.077108
	Total weight	539	377	477	477
	AVG density	5.6	4.47	5.53	5.53
	Over capacity	2	2	0	0
	route	15	15	15	15
($\boldsymbol{\alpha}$; $\boldsymbol{\beta}$; K; Kc) = (10; 10; 200; 90)	Gini coefficient	0.161905	0.2146	0.1553	0.1553
	Total weight	539	364	430	430
	AVG density	5.6	4.27	4.87	4.87
	Over capacity	2	1	0	0
	route	15	15	15	15

	Gini coefficient	0.161905	0.2570	0.1221	0.1221
	Total weight	539	365	416	416
(**α**; **β**; K; Kc) = (2.5; 2.0; 30; 10)	AVG density	5.6	4.6	4.73	4.73
	Over capacity	2	1	0	0
	route	15	15	15	15

In the experiment with $n = 21$, all road segments were filled with vehicles, but not all route selection data showed well-distributed results. For example, in the QW results, there is a road segment filled with 14 vehicles, which means there is severe congestion in this segment. Meanwhile, the QC results indicate that the QUBO model is still able to keep the number of vehicles from exceeding the road capacity, even though the vehicle queues in each segment are already quite dense (between 6 and 9 vehicles). In figure 4 there is a road segment occupancy reaching 175% of the segment's capacity, which proves that the QW model is unable to prevent the accumulation of vehicles. This accumulation has the potential to cause traffic congestion on the road. In contrast, the outcomes of QC+Amplify and QC+Neal effectively mitigate the risk of road congestion by preserving the segment's road occupancy, typically within the range of 60-80% of its allocated capacity. The vehicle route from Init becomes the one that generates the most segments of dense and potentially congested roads, with data in the third quartile reaching 100% occupancy. At this threshold occupancy levels suggest a potential severe impact on traffic flow, resulting in delays and longer travel times.

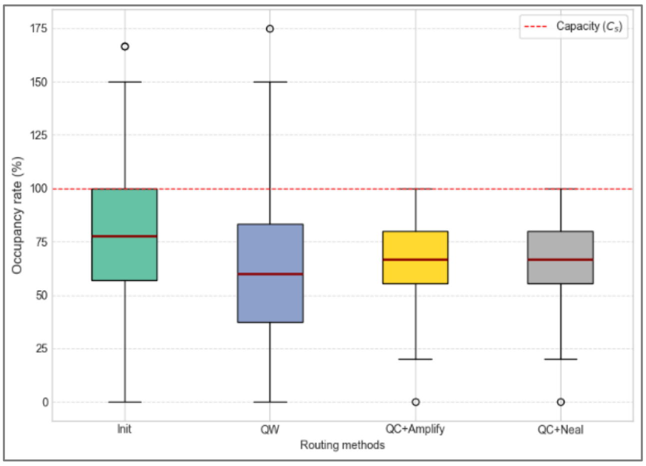

Figure 4. Comparison of Routing Methods For Route Segment Occupancy Rate Based On Capacity (C_s)

The Gini coefficient is used in this study to see how far the distribution of traffic is. We look at how many vehicles are on a network of road segments and compare the results based on how they are routed. A low Gini coefficient (near 0) means that the distribution is even, which means that cars are spread out almost evenly across all available segments. This is a scenario depicting a network where no single segment is disproportionately burdened. This condition results in low congestion and smooth traffic flow. Conversely, a high Gini coefficient (near 1) indicates severe imbalance. This means that a large number of vehicles are concentrated on only a few road segments, and there may be other segments that are underutilized. This value could be a sign of a possible statistical traffic congestion or bottleneck. According to table 3, the QW model always has the highest Gini coefficient value. This measurement means that the road use is likely to be uneven, which could cause traffic jams in real life. For instance, when the load is high (n=19), the Gini coefficient for the QW model goes up to 0.34, which means that traffic is not evenly distributed and there is a chance of congestion. In the same scenario, the QC model (Amplify or Neal) maintains a very low Gini of around 0.08, indicating a distribution that more evenly than the QW model. This is because the QW model focuses on finding routes with the smallest route cost weight, so route choices will be concentrated in segments with small weight values. The highest density tested was n = 21. The QC model excelled, achieving a Gini coefficient of just 0.0496, resulting in a distribution that was nearly even. Meanwhile, the QW and Init models struggled with much higher inequality values of 0.2370 and 0.1739, respectively.

For the QC model, the vehicle route choices generated by both annealers (Amplify and Neal) are identical. The resulting data demonstrates (see table 3) that the QUBO model created to address traffic congestion is valid for use on both the

SA and QA engines, which use the QUBO model for optimization. However, the increasing number of QUBO variables affects the performance of the SA machine. The execution time of QC+Neal (use num_reads = 1000) shows exponential growth. For small problems (n < 9), this method is faster. However, as the number of QUBO variables increases, the time required increases drastically (see figure 5). This is reasonable because Neal treats the annealing process similarly to QA on the D-Wave, which uses the number of samples as its optimization parameter. The larger the value of num_reads, the more iterations the process undergoes, and the greater the chance of obtaining the global optimal solution. On the other hand, Neal acts as an annealing simulator that works on a local CPU, unlike Amplify, which executes on its own GPU server. Therefore, the specifications of the local CPU used will have a significant impact.

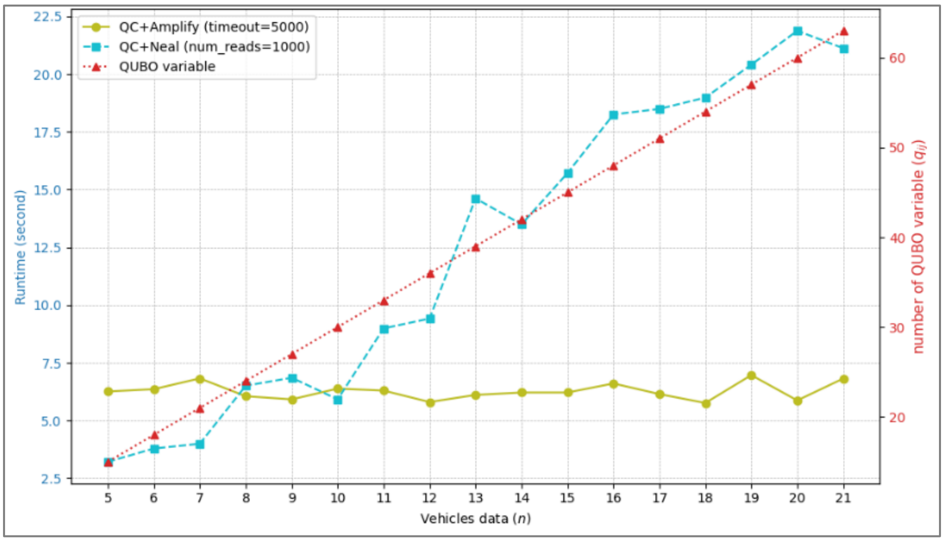

Figure 5. Runtime (seconds): Fixstars Amplify Vs D-Wave Neal Based On QUBO Variable (q_{ij}) For Each Vehicle

Unlike Fixstars Amplify, which utilizes a GPU, a timeout duration limits the computation process. A model QUBO with too many and complex variables can result in no solution being found because the execution time is limited. However, in the experiments conducted in this paper, a graph with 15 edges can still handle a vehicle count of $n = 21$ with a timeout value of 5 seconds. And because of that time execution limitation, Amplify has a relatively stable runtime duration for the optimization process.

4. Conclusion

We have demonstrated that the QUBO model with a quantum-inspired approach can solve traffic congestion problems. We address traffic congestion in this case by considering route density and imposing capacity constraints on segments. Both of these factors directly influence vehicle choices regarding routes that are almost evenly distributed across all road segments. The application of density (d_s) in this problem successfully addresses route selection without adding new congestion to the chosen segments. Meanwhile, the use of capacity (C_s) as a penalty keeps each segment from becoming congested with vehicles. However, it is also necessary to consider developing validated segment capacity values that reflect actual road conditions, such as those that take into account the length of the road. This level of density has the potential to affect congestion on road segments. The results of the experiments we conducted (see figure 4) show that the QUBO congestion model (QC) can maintain road occupancy in the range of 60-85%. Under the same conditions, the QUBO weighted segment model (QW) is able to provide route solutions with optimal weights, but road occupancy is relatively high, even reaching 175% of the required capacity. Sensitivity analysis of traffic optimization can be affected by increasing the number of vehicles or vehicle route choices that result in changes in road density levels (such as the example of total traffic data in table 1). Whether increasing the number of vehicles or changing the combination of vehicle route choices, the road density configuration (d_s) will globally change. However, such changes can be handled well by the proposed QUBO model, which is able to maintain a relatively even occupancy rate and does not exceed the capacity limit (see figure 4). Likewise, as seen in table 3, the results of the QC model route selection in all conditions show that the number of vehicles is relatively evenly distributed across all segments (with a Gini

ISSN 2723-6471

coefficient value close to 0). However, the QC model does not achieve results as satisfactory as QW in terms of maintaining total weight costs, because this condition is a trade-off against the even distribution of vehicles across.

The research we have conducted can serve as a starting point for the development of traffic management systems using the quantum annealing approach. Although there are limitations in access to quantum annealing machines, simulations using Amplify and Neal demonstrated the QUBO congestion model's validity and feasibility. Technically, the use of Neal is closer to the QA machine, where the annealing process is influenced by the number of samples in num_reads, which is probabilistic. However, the runtime on QC+Neal shows a significant increase in time as the number of vehicles increases (affecting the number of qubits). The runtime of Neal becomes slow because the computational workload runs on the local CPU, unlike QC+Amplify which executes on the GPU machine. Therefore, the timeout parameter on Amplify is purely computational on the server without considering network latency, process queuing, or result retrieval on the local computer. Thus, the use of Amplify will be suitable for simulating QUBO models with a large number of variables before being applied to quantum annealing machines. Thus, the use of Amplify would be suitable for simulating a QUBO model with a large number of variables before applying it to a quantum annealing machine. In terms of results, using Amplify and Neal yielded similar vehicle route optimization results (total weight and density). When implementing the QUBO model in real-world conditions, the choice between a commercial platform like Fixstars Amplify and open-source software like Neal presents trade-offs. However, the use of both annealers is intended to test the feasibility of the QUBO model designed for traffic congestion optimization.

This QUBO model's experimental results hold significant potential for application in real road network models. Especially in the problem of addressing congestion integrated with many parameters. The challenge is to ensure that the new route provided does not create new congestion on the selected road segment. Moreover, the implementation of multiple routes will make the cost optimization compromise increasingly complicated. Route selection is represented by a binary variable in the QUBO congestion model, which consists of n vehicles and j={1,2,3} alternative routes. This configuration requires n × j binary variables in the QUBO matrix. Consequently, increasing the number of alternative routes will significantly enlarge the problem's scale and enhance its computational complexity. The growth in the number of vehicles and route will make annealing computation more complex, and this poses a challenge for implementation on current quantum machines that require proper handling of variable qubits. The current number of 15 road segments can be further expanded in a real-world implementation without changing the QUBO model. The complexity of this QUBO model depends on the number of overlapping routes used by vehicles, creating a QUBO matrix that is increasingly dense in its quadratic elements. Such a scenario can cause the complexity to increase to O(n^2). However, if each route segment is used evenly by vehicles, minimizing overlap, the complexity can approach O(n), consistent with the linear element form of the QUBO matrix. The mathematical structure of the QUBO congestion model remains the same whether applied to our test case or a real-world map. While real-world implementations will increase the scale and computational challenges (e.g., qubit in QA), the principle of how complexity arises from route overlap remains unchanged. This understanding is an important consideration for any future large-scale implementations.

The research results obtained at this time are deliberately limited, because they only focus on the application of road density problems to overcome traffic congestion. In further development, road density data can be dynamically extracted from the application of time windows, for example, the history of morning or evening conditions that may cause some road sections to experience excessive congestion or other conditions. In addition, it is also necessary to consider the development of parameters that are in accordance with real conditions, such as different types of vehicles, heterogeneous road capacities, driver preferences (which affect driving speed), duration and lane settings caused by traffic lights, and waiting times at intersections, which are important steps toward implementation in the real world.

5. Declarations

5.1. Author Contributions

Conceptualization: T.D.T., I.J.M.E.; Methodology: T.D.T., A.B.S.; Software: I.J.M.E., R.M.; Validation: A.B.S., T.D.T.; Formal Analysis: T.D.T.; Investigation: I.J.M.E., R.M.; Resources: A.B.S., R.M.; Data Curation: I.J.M.E.;

Writing – Original Draft Preparation: T.D.T.; Writing – Review and Editing: A.B.S., R.M.; Visualization: R.M.; All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

5.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

5.4. Institutional Review Board Statement

Not applicable.

5.5. Informed Consent Statement

Not applicable.

5.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. Shikanai, M. Ohzeki, and K. Tanaka, "Quadratic unconstrained binary formulation for traffic signal optimization on real-world maps," *J. Phys. Soc. Jpn.*, vol. 94, no. 2, pp. 1–9, Feb. 2025, doi: 10.7566/JPSJ.94.024001.

[2] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, "Quantum annealing for industry applications: Introduction and review," *Rep. Prog. Phys.*, vol. 2021, no. Dec., pp. 1–35, Dec. 2021, doi: 10.1088/1361-6633/ac8c54.

[3] F. Glover, G. Kochenberger, and Y. Du, "A tutorial on formulating and using QUBO models," *arXiv*, vol. 2019, no. Nov., pp. 1–42, 2019, doi: 10.48550/arXiv.1811.11538.

[4] A. Singh, C.-Y. Lin, C.-I. Huang, and F.-P. Lin, "Quantum annealing approach for the optimal real-time traffic control using QUBO," *IEEE Softw. Eng. Comput.*, vol. 2021, no. Nov., pp. 74–81, Nov. 2021, doi: 10.1109/SNPD51163.2021.9704985.

[5] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, "Perspectives of quantum annealing: Methods and implementations," *Rep. Prog. Phys.*, vol. 83, no. 5, pp. 1–47, Mar. 2019, doi: 10.1088/1361-6633/ab85b8.

[6] S. Feld *et al.*, "A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer," *Front. ICT*, vol. 2019, no. Jan., pp. 1–14, Jan. 2019, doi: 10.3389/fict.2019.00013.

[7] N. Mori and S. Furukawa, "Quantum annealing for the adjuster routing problem," *Front. Phys.*, vol. 11, no. Mar., pp. 1–10, Mar. 2023, doi: 10.3389/fphy.2023.1129594.

[8] T. Matsumori, M. Taki, and T. Kadowaki, "Application of QUBO solver using black-box optimization to structural design for resonance avoidance," *Sci. Rep.*, vol. 12, no. 1, pp. 1–12, Dec. 2022, doi: 10.1038/s41598-022-16149-8.

[9] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Phys. Rev. E*, vol. 58, no. 5, pp. 5355–5363, Nov. 1998, doi: 10.1103/PhysRevE.58.5355.

[10] M. Ohzeki, "Breaking limitation of quantum annealer in solving optimization problems under constraints," *Sci. Rep.*, vol. 10, no. 1, pp. 1–9, Dec. 2020, doi: 10.1038/s41598-020-60022-5.

[11] A. S. Boev *et al.*, "Quantum-inspired optimization for wavelength assignment," *Front. Phys.*, vol. 10, no. Jan., pp. 1–12, Jan. 2023, doi: 10.3389/fphy.2022.1092065.

[12] A. Syrichas and A. Crispin, "Large-scale vehicle routing problems: Quantum annealing, tunings and results," *Comput. Oper. Res.*, vol. 87, no. Nov., pp. 52–62, Nov. 2017, doi: 10.1016/j.cor.2017.05.014.

[13] K. Henke, E. Pelofske, G. Kenyon, and G. Hahn, "Comparing quantum annealing and spiking neuromorphic computing for sampling binary sparse coding QUBO problems," *arXiv*, vol. 2024, no. May, pp. 1–16, May 2024.

[14] F. Neukart, G. Compostella, C. Seidel, D. von Dollen, S. Yarkoni, and B. Parney, "Traffic flow optimization using a quantum annealer," *Front. ICT*, vol. 4, no. Dec., pp. 1–9, Dec. 2017, doi: 10.3389/fict.2017.00029.

[15] T. D. Tambunan, A. B. Suksmono, I. J. M. Edward, and R. Mulyawan, "Quantum annealing for vehicle routing problem with weighted segment," *AIP Conf. Proc.*, vol. 2023, no. Jul., pp. 1–6, Jul. 2023, doi: 10.1063/5.0178362.

[16] H. Salloum, S. Zhanalin, M. Mazzara, and Y. Kholodov, "Quantum congestion-focused traffic optimization (Q-CFTO): Enhancing traffic congestion solutions with quantum annealing," *TechRxiv*, vol. 2024, no. Apr., pp. 1–12, Apr. 2024, doi: 10.36227/techrxiv.172349339.99412287/v1.

[17] D. A. Battaglia, G. E. Santoro, and E. Tosatti, "Optimization by quantum annealing: Lessons from hard satisfiability problems," *Phys. Rev. E*, vol. 71, no. 6, pp. 066707-1–066707-10, Jun. 2005, doi: 10.1103/PhysRevE.71.066707.

[18] D-Wave Systems Inc., "D-Wave documentation," *D-Wave Quantum*, vol. 2025, no. May, pp. 1–50, May 2025.

[19] A. B. Suksmono and Y. Minato, "Finding Hadamard matrices by a quantum annealing machine," *Sci. Rep.*, vol. 9, no. 1, pp. 1–9, Dec. 2019, doi: 10.1038/s41598-019-50473-w.

[20] Fixstars Corporation, "API reference: Fixstars amplify annealing engine," *Fixstars Amplify Documentation*, vol. 2024, no. Jan., pp. 1–40, Jan. 2024.

[21] 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, Jun. 2022, doi: 10.1109/ACCESS.2022.3177790.

[22] P. Debevere, M. Sugimura, and M. Parizy, "Quadratic unconstrained binary optimization for the automotive paint shop problem," *IEEE Access*, vol. 11, pp. 97769–97777, Aug. 2023, doi: 10.1109/ACCESS.2023.3313102.

[23] F. A. Quinton, P. A. S. Myhr, M. Barani, P. C. del Granado, and H. Zhang, "Quantum annealing versus classical solvers: Applications, challenges and limitations for optimisation problems," *Sci. Rep.*, vol. 2024, no. Sep., pp. 1–18, Sep. 2024, doi: 10.1038/s41598-025-96220-2.