Metaheuristic Algorithms with Applications

www.maa.reapress.com.

Metaheur. Algor. Appl. Vol. 1, No. 3 (2024) 227–248.

Paper Type: Original Article

Graph-Based Metaheuristic Optimization for Real-Time Urban Traffic Signal Coordination

Hassan Boussouf[1,*], Nguyen Minh Duc[2]

[1] Department of Computer Science, Faculty of Mathematics and Computer Science, University of Algiers 1, Algiers, Algeria; h.boussouf@univ-alger.dz.

[2] Department of Information Technology, Vietnam National University, Hanoi, Vietnam; nmduc@vnu.edu.vn.

Citation:

Received: 17 March 2024 *Revised: 03 May 2024* *Accepted: 07 July 2024*	Boussouf, H., & Minh Duc, N. (2024). Graph-based metaheuristic optimization for real-time urban traffic signal coordination. *Metaheuristic algorithms with applications, 1*(3), 227-248.

Abstract

Urban traffic congestion remains one of the most pressing challenges confronting smart city development worldwide, imposing substantial economic costs, environmental degradation, and diminished quality of life for urban residents. Existing traffic signal control paradigms, including fixed-time plans, actuated controllers, and adaptive systems such as Split Cycle Offset Optimisation Technique (SCOOT) and Sydney Coordinated Adaptive Traffic System (SCATS), exhibit significant limitations in network-wide coordination under dynamically varying demand conditions. More recently, Reinforcement Learning (RL)-based approaches have demonstrated promise at isolated intersections but face persistent challenges in scalability, sample efficiency, transferability, and real-time multi-intersection coordination. This paper presents Graph-Based Metaheuristic For Traffic Signal Coordination (GRAPH-TSC), a novel algorithm that models the urban traffic network as a directed graph and employs a graph-aware Ant Colony Optimization (ACO) variant hybridized with local search to jointly optimize cycle lengths, green splits, and offsets across all intersections in real time. GRAPH-TSC introduces four key innovations: 1) a Graph Attention Network (GAT)-based surrogate model that replaces computationally expensive microsimulation for rapid fitness evaluation during optimization, 2) a pheromone diffusion mechanism that propagates pheromone updates along the traffic graph in proportion to inter-intersection traffic flow dependencies, naturally encoding green wave coordination, 3) a hierarchical network decomposition strategy that partitions large networks into overlapping subnetworks for scalable optimization with boundary coordination via shared pheromone, and 4) an adaptive re-optimization trigger based on Cumulative Sum (CUSUM) change detection that initiates re-optimization only upon detected traffic pattern shifts. GRAPH-TSC was evaluated on SUMO-simulated networks modeled after real urban areas in Algiers, Algeria (42 intersections) and Hanoi, Vietnam (67 intersections), as well as synthetic grid networks of up to 100 intersections. Experimental results demonstrate that GRAPH-TSC achieves an 18–27% reduction in average vehicle delay, a 14–22% improvement in network throughput, and an 11–19% reduction in CO_2 emissions compared to Webster's fixed-time, SCATS-like adaptive, standard ACO, Genetic Algorithm (GA), and Deep Q-Network (DQN)-based methods. Re-optimization is completed within 3 seconds for the 67-intersection Hanoi network, confirming the algorithm's real-time applicability for operational deployment in smart city traffic management systems.

Keywords: Traffic signal coordination, Metaheuristic optimization, Ant colony optimization, Graph neural network, Smart cities, Urban traffic management, Real-time optimization.

1 | Introduction

Urban traffic congestion constitutes a pervasive and escalating global challenge, with profound implications for economic productivity, public health, environmental sustainability, and social well-being. According to

Corresponding Author: h.boussouf@univ-alger.dz

https://doi.org/10.48313/maa.v1i3.99

estimates from the Texas A&M Transportation Institute, traffic congestion in the United States alone costs commuters over 8.7 billion hours of additional travel time and 3.5 billion gallons of excess fuel consumption annually Schrank et al. [1]. Comparable magnitudes of loss are observed across rapidly urbanizing regions in Africa, Southeast Asia, and the Middle East, where infrastructure development has often lagged behind the pace of population growth and motorization Cervero [2]. In Algiers, the capital of Algeria, chronic traffic congestion in the central business district imposes daily delays of 30 to 60 minutes on commuters traversing distances as short as five kilometers. In Hanoi, Vietnam, the complex interplay of motorcycle-dominant mixed traffic and an irregular road network produces uniquely challenging congestion dynamics that conventional signal control methods are ill-equipped to address [3].

Traffic signal control has long been recognized as one of the most effective levers for mitigating urban congestion without the capital-intensive requirement of expanding physical road infrastructure. The historical evolution of signal control methods can be traced through several generations. First-generation fixed-time methods, exemplified by the seminal work of, compute optimal cycle lengths and green splits based on observed traffic volumes and saturation flow rates. While computationally straightforward and robust, fixed-time plans are inherently inflexible, unable to respond to temporal fluctuations in demand or the occurrence of non-recurrent events such as incidents and special events. Second-generation actuated controllers utilize detector data to extend or terminate phases in real time, offering improved responsiveness at individual intersections but providing limited capacity for network-wide coordination [4].

Third-generation adaptive signal control systems represent a significant advancement in coordinated, demand-responsive signal management. The Split Cycle Offset Optimisation Technique (SCOOT), developed by Hunt et al. [5] at the Transport Research Laboratory in the United Kingdom, employs online optimization of signal parameters using upstream detector data and cyclic flow profiles. The Sydney Coordinated Adaptive Traffic System (SCATS), developed by Lowrie [6] in Australia, implements a hierarchical control architecture with plan selection from a library of pre-computed timing plans based on measured traffic conditions. While SCOOT and SCATS have been widely deployed and have demonstrated measurable congestion reductions, their effectiveness diminishes under highly variable demand, oversaturated conditions, and complex network topologies where the assumptions underlying their optimization models are violated Stevanovic [7]. Other notable adaptive systems include Real-time Hierarchical, Optimized, Distributed and Effective system (RHODES) Head et al. [8] and InSync Rhythm Engineering, each with distinct architectural philosophies and deployment experiences.

The past decade has witnessed a surge of research interest in applying Reinforcement Learning (RL) methods to traffic signal control. Early work demonstrated the viability of Deep Q-Network (DQN) for learning effective signal policies at isolated intersections Li et al. [9]. Subsequent efforts advanced multi-agent RL formulations capable of coordinating multiple intersections, including Multi-Agent RL for Integrated Network (MARLIN) El-Tantawy et al. [10], Multi-Agent Actor-Critic (MA2C) Chu et al. [11], and graph-based approaches such as CoLight Wei et al. [12], PressLight Wei et al. [13], AttendLight Oroojlooy et al. [14], and DuaLight Lu et al. [15]. Although RL-based methods have shown promising performance in simulation environments, several fundamental challenges impede their practical deployment: 1) extensive training requirements and sample inefficiency, necessitating millions of simulation episodes, 2) poor transferability across networks with different topologies or demand characteristics, 4) the sim-to-real gap, whereby policies trained in simulation may fail to generalize to real-world conditions, and 5) the opacity of learned policies, which poses challenges for transportation engineers seeking to understand, validate, and trust control decisions [16].

Metaheuristic optimization methods, including Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), Differential Evolution (DE), and Ant Colony Optimization (ACO), offer a complementary paradigm for traffic signal optimization. These methods are population-based, derivative-free, and capable of navigating complex, multimodal search spaces. Ceylan and Bell [17] demonstrated the efficacy of GA for optimizing signal timing parameters across small networks. Park et al. [18] applied GA to coordinate signal offsets along arterial corridors. PSO variants have been employed for cycle length and offset optimization with promising results [19]. ACO, originally inspired by the foraging behavior of ant colonies [20], has been applied to traffic signal optimization and routing with encouraging outcomes [21]. However, the majority of metaheuristic applications in traffic signal control have been limited to isolated intersections, short arterial corridors, or small networks, and have relied on computationally expensive simulation-based fitness evaluation that precludes real-time application [22].

The emergence of Graph Neural Networks (GNNs) as a powerful paradigm for learning over graph-structured data provides a natural bridge between the graph topology of urban traffic networks and data-driven modeling. GNN architectures, including Graph Convolutional Networks Kipf and Welling [23], Graph Attention Network (GAT) Veličković et al. [24], and spatio-temporal variants such as Diffusion Convolutional Recurrent Neural Network (DCRNN) Li et al. [25] and Spatio-Temporal Graph Convolutional Network (STGCN) Yu et al. [26], have demonstrated state-of-the-art performance in traffic flow prediction and traffic state estimation. The capacity of GNNs to capture spatial dependencies along road network graphs, combined with their computational efficiency at inference time, renders them compelling candidates for constructing surrogate models that can approximate the input–output behavior of computationally expensive traffic microsimulation models.

Despite these advances across the domains of metaheuristic optimization, GNN-based traffic modeling, and adaptive signal control, a critical research gap persists: no existing method integrates graph-aware metaheuristic optimization with GNN surrogate models to enable real-time, network-wide Graph-Based Metaheuristic For Traffic Signal Coordination (GRAPH-TSC). This paper addresses that gap by proposing GRAPH-TSC, a novel algorithm that synergistically combines these capabilities. The principal contributions of this paper are as follows:

A novel graph-aware ACO algorithm for network-wide GRAPH-TSC that traverses the traffic graph during solution construction, inherently capturing spatial dependencies among intersections.

A pheromone diffusion mechanism that propagates pheromone information along the traffic graph in proportion to traffic flow dependencies between adjacent intersections, naturally encoding green wave coordination effects.

A GAT-based surrogate model that replaces computationally expensive microsimulation during metaheuristic fitness evaluation, enabling optimization within seconds rather than hours.

A hierarchical network decomposition strategy based on spectral clustering that partitions large networks into overlapping subnetworks for scalable optimization with boundary coordination via shared pheromone.

An adaptive re-optimization trigger based on Cumulative Sum (CUSUM) change detection that initiates re-optimization only upon detected traffic pattern shifts, avoiding unnecessary computational overhead during stable traffic conditions.

The remainder of this paper is organized as follows. Section 2 reviews related work on traffic signal control, RL-based approaches, metaheuristic optimization, and GNN-based traffic modeling. Section 3 presents the detailed methodology of GRAPH-TSC. Section 4 describes the experimental setup, including test networks, demand scenarios, and comparison methods. Section 5 presents the experimental results and analysis. Section 6 provides a critical discussion of findings, limitations, and future directions. Section 7 concludes the paper.

2 | Related Work

2.1 | Traffic Signal Control Methods

The design and operation of traffic signal control systems has been a subject of active research for over seven decades, yielding a rich taxonomy of methods that can be broadly classified into fixed-time, actuated, and adaptive categories. Fixed-time signal control, the most foundational approach, computes signal timing parameters cycle length, green splits, and offsets based on historical or surveyed traffic volumes. Webster's (1958) formula for optimal cycle length remains a standard reference, providing a closed-form expression that minimizes total intersection delay as a function of critical flow ratios and lost time. The Highway Capacity Manual (HCM) [27] further codified procedures for fixed-time signal design, incorporating level-of-service analysis and progression quality measures. Bandwidth maximization methods, notably the MAXBAND formulation [28] and its extensions, MULTIBAND [29] and Asymmetric Multi Band(AM-BAND) [30], focus on maximizing the width of green bands along arterials to facilitate coordinated vehicle progression. While fixed-time and bandwidth methods provide reliable baseline performance, they cannot adapt to real-time demand fluctuations, incident conditions, or non-recurrent congestion events.

Actuated signal control employs vehicle detectors (typically inductive loops or video detection) to dynamically extend or terminate signal phases based on real-time demand at the intersection level. The National Electrical Manufacturers Association (NEMA) standards define the operational framework for actuated controllers, including gap-out and max-out logic, phase recalls, and coordination modes NEMA [31]. While actuated control improves intersection-level efficiency relative to fixed-time plans, its responsiveness is inherently local, and coordination among actuated intersections is typically achieved through background cycle and offset constraints that may conflict with the actuated phase logic.

Adaptive signal control systems represent the most sophisticated operational paradigm, employing online optimization algorithms that continuously adjust signal parameters in response to measured or predicted traffic conditions. SCOOT Hunt et al. [5] optimizes splits, offsets, and cycle lengths through incremental adjustments every few seconds, guided by cyclic flow profiles predicted from upstream detector data. SCATS Lowrie [6] implements a hierarchical control architecture, selecting timing plans from a pre-computed library based on the degree of saturation measured at each intersection. RHODES Head et al. [8] utilizes a hierarchical prediction-based approach with dynamic programming for phase sequence optimization. More recently, Model Predictive Control (MPC) formulations have been applied to traffic signal control, utilizing rolling-horizon optimization based on traffic-state prediction models by Aboudolas et al. [32] and Lin et al. [33]. The MaxPressure control policy Varaiya [34] provides a decentralized, provably stable control strategy that selects phases to maximize the pressure (difference in queue lengths) at each intersection, achieving throughput optimality under certain assumptions.

2.2 | Reinforcement Learning-Based Traffic Signal Control

The application of RL to traffic signal control has expanded rapidly since the mid-2010s, propelled by advances in deep RL and the availability of high-fidelity traffic simulators. Li et al. [9] demonstrated the viability of DQN for learning phase-switching policies at isolated intersections, using detector-derived state representations and reward functions based on cumulative delay. Subsequent work extended RL approaches to multi-intersection settings, confronting the combinatorial explosion of joint action spaces and the challenges of multi-agent coordination. El-Tantawy et al. [10] proposed MARLIN for adaptive traffic signal controllers, employing independent learners with shared reward structures. Chu et al. [11] introduced MA2C, incorporating spatial discount factors and mean-field approximations to stabilize learning in networked settings.

Graph-based RL approaches have emerged as a promising direction for capturing the topological structure of traffic networks. Wei et al. [12] proposed CoLight, which employs GAT to enable communication among intersection agents, achieving effective coordination without requiring explicit message-passing protocols. PressLight: Wei et al. [13] integrate the maxpressure concept into RL reward design, leveraging traffic theory to guide the learning process. Oroojlooy et al. [14] developed AttendLight, using attention mechanisms to weight contributions from neighboring intersections. Lu et al. [15] introduced DuaLight, a dual-pathway architecture that processes both intersection-level and network-level features. Despite their achievements in simulation, RL-based methods face persistent challenges: 1) sample efficiency remains a concern, with state-of-the-art methods requiring millions of training episodes, 2) policies trained in one network topology or demand regime exhibit poor transferability to different settings, 3) the sim-to-real gap poses risks when deploying learned policies in real traffic environments, and 4) the black-box nature of neural network policies limits interpretability and complicates the regulatory approval processes required for deployment in public infrastructure [16], [35].

2.3 | Metaheuristic Optimization for Traffic Signals

Metaheuristic optimization methods have been applied to various facets of traffic signal design and operation. GA, among the earliest and most widely applied metaheuristics for signal optimization, was employed by Ceylan and Bell [17] to simultaneously optimize cycle lengths, green splits, and offsets for small networks, using microsimulation-based fitness evaluation. Foy et al. [36] demonstrated GA-based optimization for signal timing at isolated intersections. Park et al. [18] applied GA to optimize offsets along arterial corridors for coordinated progression. More recently, multi-objective evolutionary algorithms, particularly Non-dominated Sorting GA-II (NSGA-II) [37], have been applied to simultaneously optimize delay, emissions, and fuel consumption in signal timing design [7].

PSO has been applied to signal timing by Garcia et al. [19], who demonstrated competitive performance with GAs while requiring fewer function evaluations. Variants have been explored for signal parameter optimization, with promising convergence behavior [38]. ACO, inspired by the pheromone-based communication of foraging ants Dorigo [20], has been applied primarily to traffic routing problems but has also been adapted for signal timing optimization. D'Acierno et al. [21] employed ACO for jointly optimizing signal timing and route guidance in urban networks. Khamis and Gomaa [39] proposed an ACO variant for adaptive traffic signal control at isolated intersections.

A persistent limitation across metaheuristic approaches for traffic signal optimization is their reliance on microsimulation for fitness evaluation. Each candidate solution requires a full simulation run, and the stochastic nature of microsimulation necessitates multiple replications per evaluation. For a typical ACO or GA run involving thousands of evaluations, the computational burden renders real-time application infeasible. This limitation motivates the integration of surrogate models to accelerate fitness evaluation, as proposed in the present work.

2.4 | Graph Neural Networks for Traffic

Graph neural networks have emerged as a dominant paradigm for traffic prediction and state estimation, owing to their natural capacity to model the spatial dependencies inherent in road network graphs. Li et al. [25] proposed the DCRNN, which models traffic flow as a diffusion process on a directed graph and employs a sequence-to-sequence architecture for multi-step prediction. Yu et al. [26] introduced the STGCN, combining graph convolutions with temporal gated convolutions for efficient spatio-temporal feature extraction. Wu et al. [40] proposed Graph WaveNet, which adaptively learns the graph structure from data, eliminating the need for a pre-defined adjacency matrix. Zheng et al. [41] developed the Graph Multi-Attention Network (GMAN), which incorporates spatial and temporal attention mechanisms for long-range traffic prediction.

Beyond traffic prediction, GNNs have been employed as surrogate models for traffic simulation, offering rapid approximation of simulation outputs at a fraction of the computational cost. Liang et al. [42] demonstrated the use of GNN-based surrogate models for traffic assignment, achieving substantial speed-ups over traditional solvers while maintaining acceptable accuracy. Narayanan et al. [43] employed GNNs to approximate the outputs of microsimulation models for intersection-level performance evaluation, reporting speed-up factors exceeding 10,000× with Mean Absolute Errors (MAE) below 10%. These results motivate the use of GNN-based surrogates as fitness evaluators within metaheuristic optimization loops, as proposed in GRAPH-TSC.

3 | Methodology

3.1 | Problem Formulation

We model the urban traffic network as a directed graph $G = (V, E)$, where the vertex set $V = \{v_1, v_2, \ldots, v_N\}$ represents the N signalized intersections in the network, and the edge set $E \subseteq V \times V$ represents directed road links connecting adjacent intersections. Each edge $e = (v_i, v_j) \in E$ is characterized by attributes including link length l_e, free-flow speed u_e, number of lanes n_e, and saturation flow rate s_e. For each intersection $v_i \in V$, the signal timing is characterized by three classes of decision variables:

Cycle length: $C_i \in [C_{min}, C_{max}]$, typically [60, 180] seconds.

Green split vector: $g_i = (g_{i1}, g_{i2}, \ldots, g_{iP_i})$ for P_i phases, subject to the constraint $\Sigma_{p=1}^{P_i} g_{ip} = C_i - L_i$, where L_i denotes the total lost time per cycle at intersection i.

Offset: $O_i \in [0, C_i)$ representing the temporal displacement of the green onset of a reference phase relative to a system-wide reference time.

The complete decision vector for the network is $x = (C_1, g_1, O_1, \ldots, C_N, g_N, O_N)$. The optimization objective is to minimize a network performance index J defined as a weighted sum of traffic performance measures across all links:

$$J(x) = \Sigma e \in E\, we \cdot [\alpha \cdot de(x) + \beta \cdot se(x) + \gamma \cdot \varepsilon e(x)], \quad (1)$$

where $d_e(x)$ is the average vehicle delay on link e (seconds per vehicle), $s_e(x)$ is the average number of stops per vehicle on link e, $\varepsilon_e(x)$ is the emission index on link e (grams CO_2 per kilometer), w_e is a flow-weighted importance coefficient proportional to the demand volume on link e, and α, β, γ are user-specified weights reflecting the relative importance of delay, stops, and emissions (default: $\alpha = 0.6$, $\beta = 0.2$, $\gamma = 0.2$).

The optimization is subject to the following constraints:

$$gip \geq gmin, p \quad \forall\, i \in V, p = 1, \ldots, Pi, \quad (2)$$

$$gip \geq gped, p \quad \forall \text{ pedestrian phases } p, \quad (3)$$

$$Ci \leq Cmax \quad \forall\, i \in V, \quad (4)$$

where $g_{min,p}$ denotes the minimum green time for phase p (typically 7–15 seconds depending on approach geometry), and $g_{ped,p}$ denotes the minimum pedestrian clearance requirement for phase p. Phase sequence constraints are treated as fixed and pre-specified for each intersection.

3.2 | Graph Neural Network-Based Surrogate Model

A critical enabler of real-time optimization in GRAPH-TSC is the replacement of computationally expensive microsimulation with a GAT-based surrogate model that rapidly predicts network-wide traffic performance metrics given signal timing parameters and traffic demand as inputs. The surrogate model architecture is described in this section.

Graph representation. The traffic network graph $G = (V, E)$ is endowed with node feature vectors and edge feature vectors. For each intersection node v_i, the node feature vector $h_i^{(0)} \in \mathbb{R}^{d_n}$ concatenates: 1) normalized signal timing parameters (cycle length, green splits, offset), 2) current demand indicators (approach volumes per movement), and 3) geometric features (number of approaches, number of lanes per approach). For each directed edge $e = (v_i, v_j)$, the edge feature vector $f_e \in \mathbb{R}^{d_e}$ encodes link capacity, link length, free-flow speed, and speed limit. In our implementation, $d_n = 24$ and $d_e = 4$.

Architecture. The surrogate model employs a three-layer GAT architecture Veličković et al. [24] with multi-head attention. In each GAT layer, the attention mechanism computes coefficients that determine the importance of neighboring node features for updating each node's representation:

$$\alpha ij(k) = \text{softmax}j(\text{LeakyReLU}(a(k)T\ [W(k)hi \parallel W(k)hj \parallel We(k)fij])), \quad (5)$$

$$hi(l+1) = \|k = 1K\ \sigma(\Sigma j \in \mathcal{N}(i)\ \alpha ij(k)\ W(k)hj(l)), \quad (6)$$

where K = 4 attention heads are used, ‖ denotes concatenation (replaced by averaging in the final layer), $W^{(k)}$ and $a^{(k)}$ are learnable parameters for attention head k, $W_e^{(k)}$ projects edge features into the attention computation, and $\mathcal{N}(i)$ denotes the set of neighbors of node i in the traffic graph. The three GAT layers use hidden dimensions of 64, 64, and 32, respectively, with Exponential Linear Unit (ELU) activation functions and dropout (rate 0.1) for regularization.

Output layer. The final node representations are passed through a two-layer MLP (multilayer perceptron) with dimensions 32 → 16 → 3, predicting three per-link performance metrics: average delay d_e, average number of stops s_e, and emission index ε_e. The network performance index J is then computed from these predictions using *Eq. (1)*.

Training. The surrogate model is trained on a dataset of 50,000 samples generated by running the SUMO microsimulator Lopez et al. [44] with systematically varied signal timing parameters and demand patterns. Latin hypercube sampling is used to ensure uniform coverage of the parameter space. The training objective is to minimize Mean Squared Error (MSE) between surrogate predictions and simulation outputs. The model is trained using the Adam optimizer [45] with an initial learning rate of 0.001, a batch size of 256, and early stopping based on validation loss (10% holdout). Training requires approximately 8 minutes on an NVIDIA RTX 3080 Graphics Processing Unit (GPU).

Inference speed and accuracy. At inference time, the surrogate model evaluates a complete signal timing plan in approximately 2 ms, compared to approximately 30 seconds for a full SUMO simulation run. This speed difference represents a speed-up factor of approximately 15,000×, which is the critical enabler of real-time metaheuristic optimization. The surrogate is periodically retrained every 30 minutes of simulated time with fresh simulation data to maintain accuracy, with retraining requiring approximately 45 seconds using warm-starting from the previous model parameters. Validation experiments demonstrate that the surrogate achieves a MAE of 2.3 seconds for delay prediction, corresponding to a relative error of 7.1% compared to full SUMO simulation (see Section 5.4 for detailed analysis).

3.3 | Graph-Aware Ant Colony Optimization

GRAPH-TSC employs a modified ACO algorithm that is specifically designed to exploit the graph structure of the traffic network during solution construction and pheromone management. Unlike standard ACO formulations for combinatorial optimization, which operate on abstract solution components, GRAPH-TSC's ants physically traverse the traffic graph, constructing signal timing plans in a spatially coherent manner that naturally captures inter-intersection dependencies.

Solution construction. Each ant constructs a complete network-wide signal timing plan by traversing the traffic graph starting from a randomly selected seed intersection. From the seed intersection, the ant selects timing parameters (cycle length, green split, offset) for that intersection based on pheromone and heuristic information, then moves to an unvisited adjacent intersection, selects its parameters, and continues until all intersections have been assigned timing parameters. The traversal order ensures that when an ant selects the offset for intersection v_j, it has already determined the timing parameters of at least one adjacent intersection, allowing the offset selection to be informed by the timing context of the local neighborhood. This traversal strategy is a fundamental departure from standard metaheuristic approaches that treat each intersection's parameters independently.

Pheromone representation. The pheromone model is defined over discretized parameter bins. The continuous parameter space for each intersection is discretized into B = 20 bins per parameter dimension. Pheromone values are maintained as $\tau(i, b_c, b_g, b_o)$, representing the desirability of assigning cycle length bin b_c, green split bin vector b_g, and offset bin b_o to intersection i. To manage the combinatorial explosion of joint bins, we factorize the pheromone into separate components: $\tau_C(i, b_c)$, $\tau_G(i, b_g)$, and $\tau_O(i, b_o \mid j, b_{o,j})$, where the offset pheromone is conditioned on the offset of the previously visited adjacent intersection j to capture coordination effects.

Heuristic information. The heuristic information η provides problem-specific guidance to the ant's construction process. For cycle length, the heuristic is derived from Webster's (1958) optimal cycle formula:

$$C_{opt,i} = (1.5\, L_i + 5) / (1 - Y_i)\ (7), \qquad \textbf{(7)}$$

where L_i is total lost time and $Y_i = \Sigma_p\, y_{ip}$ is the sum of critical flow ratios across phases at intersection i. The heuristic value for cycle length bin b_c is inversely proportional to the distance from $C_{opt,i}$: $\eta_C(i, b_c) = 1 / (1 + |C(b_c) - C_{opt,i}|)$. For green splits, the heuristic is derived from Webster's equi-saturation principle: $g_{ip}^* = y_{ip} / Y_i \cdot (C_i - L_i)$. For offsets, the heuristic encodes the ideal offset for green wave progression based on link travel time between intersections: $O_j^* = O_i + l_{ij} / u_{ij}$ (mod C).

Transition probability. The probability that an ant selects parameter bin b for parameter type t at intersection i is given by:

$$p(i,b) = [\tau t(i,b)]\alpha \cdot [\eta t(i,b)]\beta / \Sigma b' \, [\tau t(i,b')]\alpha \cdot [\eta t(i,b')]\beta, \qquad \textbf{(8)}$$

where α and β control the relative influence of pheromone and heuristic information, respectively (default: $\alpha = 1.0$, $\beta = 2.0$).

Pheromone update with diffusion. After all ants complete their tours in each iteration, pheromone is updated using the standard evaporation-deposit rule, augmented with a novel pheromone diffusion mechanism. First, pheromone evaporation is applied:

$$\tau t(i,b) \leftarrow (1 - \rho) \cdot \tau t(i,b), \qquad \textbf{(9)}$$

where $\varrho = 0.1$ is the evaporation rate. Next, the top-K ants (ranked by fitness) deposit pheromone on the parameter bins they selected. The best ant deposits additional pheromone proportional to its fitness advantage. Following the deposit, the pheromone diffusion mechanism propagates pheromone information along the traffic graph:

$$\tau O(j, bo) \leftarrow \tau O(j, bo) + \rho diff \cdot (qij / capij) \cdot \Delta\tau O(i, bo, i), \qquad \textbf{(10)}$$

where $\varrho_{diff} = 0.3$ is the diffusion rate, q_{ij} / cap_{ij} is the volume-to-capacity ratio on the link from intersection i to intersection j (serving as a flow-dependent diffusion weight), and $\Delta\tau_O(i, b_{o,i})$ is the pheromone deposited at intersection i for offset bin $b_{o,i}$. The offset bin b_o at intersection j that receives the diffused pheromone is the bin corresponding to the ideal green wave offset: $O_j = O_i + l_{ij} / u_{ij}$ (mod C). This mechanism naturally propagates "green wave" coordination signals through the network. When a timing plan achieves good performance at intersection i, the associated offset information diffuses to adjacent intersections weighted by traffic flow intensity, encouraging future ants to select coordinated offsets along high-flow corridors.

The complete GRAPH-TSC main loop is presented in Algorithm 1.

Algorithm 1 (GRAPH-traffic signal control main loop).

Input: Traffic graph G = (V, E), current demands D, GNN surrogate model S Output: Optimized signal timing plan x* Parameters: M (ants), Imax (max iterations), α, β, ϱ, ϱdiff, B (bins) 1: Initialize pheromone τt(i, b) ← τ0 for all i, t, b 2: Compute heuristic information η using Webster's formulas (Eq. 7) 3: x* ← ∅; J* ← +∞ 4: for iteration = 1 to Imax do 5: for ant m = 1 to M do 6: Select seed intersection vseed uniformly at random 7: Initialize traversal queue Q ← {vseed}; visited ← ∅ 8: while Q ≠ ∅ do 9: vi ← Dequeue(Q); visited ← visited ∪ {vi} 10: Select Ci using Eq. (8) with τC and ηC 11: Select gi using Eq. (8) with τG and ηG 12: Select Oi using Eq. (8) with τO and ηO 13: Enforce constraints (Eqs. 2–4); repair if violated 14: Enqueue unvisited neighbors of vi into Q 15: end while 16: xm ← assembled timing plan for all intersections 17: Jm ← S.evaluate(xm, D) // GNN surrogate evaluation (~2 ms) 18: end for 19: Rank ants by fitness Jm 20: if min(Jm) < J* then 21: x* ← xbest; J* ← min(Jm) 22: end if 23: Apply pheromone evaporation (Eq. 9) 24: Deposit pheromone from top-K ants 25: Apply pheromone diffusion (Eq. 10) along traffic graph 26: if convergence criterion met then break 27: end for 28: Apply local search (Nelder-Mead) on offsets of x* (Section 3.6) 29: return x*

3.4 | Hierarchical Network Decomposition

For large networks containing more than approximately 50 intersections, the dimensionality of the parameter space and the complexity of inter-intersection dependencies render monolithic optimization computationally challenging, even with the GNN surrogate model. GRAPH-TSC addresses this scalability challenge through a hierarchical network decomposition strategy that partitions the traffic graph into overlapping subnetworks, optimizes each subnetwork independently, and coordinates boundary intersections through shared pheromone.

Graph partitioning. The traffic graph is partitioned using spectral clustering on the traffic flow-weighted adjacency matrix. Let $A \in \mathbb{R}^{N \times N}$ be the adjacency matrix where $A_{ij} = q_{ij}$ (traffic flow from intersection i to j). The normalized graph Laplacian $L = I - D^{-1/2}AD^{-1/2}$ is computed, and spectral clustering is applied to the eigenvectors corresponding to the K smallest non-zero eigenvalues to partition V into K subsets $\{V_1, V_2, \ldots, V_K\}$. The number of clusters, K, is determined such that each subnetwork contains 15–25 intersections. To ensure coordination across subnetwork boundaries, each subnetwork is expanded to include a 3–5 intersection overlap zone with adjacent subnetworks. Intersections in the overlap zone are optimized by both adjacent subnetworks, providing shared context for boundary coordination.

Hierarchical optimization. The hierarchical optimization proceeds iteratively. In each hierarchical iteration, ACO is run independently on each subnetwork (potentially in parallel on multi-core hardware). After each subnetwork's ACO run completes, the pheromone values for intersections in the overlap zones are averaged across the subnetworks that share them. The updated boundary pheromone is then used as input for the next hierarchical iteration. This process repeats until convergence, typically requiring 3–5 hierarchical iterations. The procedure is formalized in Algorithm 2.

Algorithm 2 (Hierarchical network decomposition and coordination).

Input: Traffic graph G = (V, E), current demands D, GNN surrogate S Output: Network-wide optimized signal timing plan x* 1: Compute flow-weighted adjacency matrix A from current demands 2: Apply spectral clustering to partition V into K subsets {V1, …, VK} 3: Expand each Vk with 3–5 intersection overlap from adjacent partitions 4: Initialize boundary pheromone τboundary uniformly 5: for h = 1 to Hmax do // hierarchical iterations, Hmax = 5 6: for k = 1 to K do // parallelizable 7: Inject τboundary into overlap zone pheromone of subnetwork k 8: xk* ← GRAPH-TSC(G[Vk], D, S) // Algorithm 1 on subnetwork 9: end for 10: for each overlap intersection vo shared by subnetworks k1, k2 do 11: τboundary(vo) ← average of τk1(vo) and τk2(vo) 12: end for 13: if boundary pheromone change < εconv then break 14: end for 15: x* ← Merge subnetwork solutions {x1*, …, xK*} 16: (overlap intersections: use weighted average of subnetwork solutions) 17: return x*

3.5 | Adaptive Re-Optimization Trigger

In operational deployment, continuous re-optimization on a fixed schedule is both computationally wasteful and potentially disruptive. GRAPH-TSC employs an adaptive re-optimization trigger based on the CUSUM change detection algorithm (Page, 1954) to detect statistically significant shifts in traffic patterns and initiate re-optimization only when warranted.

Let d(t) denote the network-average delay measured at time step t, and let μ denote the expected delay under the current signal plan (predicted by the GNN surrogate). The CUSUM statistic S(t) is updated as:

$$S(t) = \max(0, S(t-1) + (d(t) - \mu) - k), \tag{11}$$

where k is a slack parameter (set to 0.5 standard deviations of historical delay). Re-optimization is triggered when S(t) exceeds a threshold h, indicating a sustained increase in delay beyond the expected level. The threshold h is calibrated to achieve a false alarm rate of less than 5% during stable traffic conditions, determined empirically from validation data. Upon triggering, the CUSUM statistic is reset, current traffic demands are re-estimated from detector data, the GNN surrogate is queried with current conditions, and GRAPH-TSC is executed to generate a new optimized signal plan.

3.6 | Local Search Enhancement

After the ACO algorithm converges, the quality of the best solution is further refined through local search. Specifically, the Nelder-Mead simplex algorithm [46] is applied to the continuous offset parameters of the top-K solutions identified by ACO. The cycle lengths and green splits are held fixed during local search, as offsets have the most direct influence on inter-intersection coordination quality and can be treated as continuous variables without constraint complications. The local search is limited to a budget of 100 function evaluations per solution (using the GNN surrogate for evaluation), requiring approximately 0.2 seconds of additional computation time. This hybrid ACO-local search approach combines the global exploration capability of ACO with the local exploitation capability of the simplex method, yielding consistently improved offset coordination quality.

3.7 | Computational Complexity

The overall computational complexity of the GRAPH-TSC algorithm can be characterized as follows. In the inner ACO loop, each ant visits all N intersections and selects parameters from B bins across P parameter dimensions. With M ants and I iterations, the solution construction phase requires $O(I \cdot M \cdot N \cdot P \cdot B)$ operations. Each GNN surrogate evaluation requires $O(N \cdot d^2 \cdot L)$ operations, where d is the hidden dimension, and L is the number of GAT layers. The pheromone diffusion step requires $O(|E| \cdot B)$ operations per iteration. For the hierarchical decomposition with K subnetworks and H hierarchical iterations, the total complexity is $O(H \cdot K \cdot I \cdot M \cdot (N/K) \cdot P \cdot B)$, with subnetworks optimizable in parallel.

With the default parameters (M = 50 ants, I = 200 iterations, B = 20 bins, K = 3–4 subnetworks, H = 5 hierarchical iterations) and GNN surrogate evaluation at approximately 2 ms per evaluation, the total re-optimization time for the 67-intersection Hanoi network is under 3 seconds on a workstation equipped with an Intel Core i7-12700K CPU and NVIDIA RTX 3080 GPU. The resulting computational performance confirms the feasibility of real-time deployment.

4 | Experimental Setup

4.1 | Simulation Platform

All experiments were conducted using the SUMO microsimulator version 1.18 Lopez et al. [44]. SUMO is an open-source, microscopic, multi-modal traffic simulation platform that models individual vehicle movements based on car-following models (Krauss model), lane-changing models, and intersection control logic. The Traffic Control Interface (TraCI) was used for real-time communication between the optimization algorithm and the running simulation, enabling dynamic modification of signal timing plans during simulation. Vehicle emissions were computed using SUMO's built-in Handbook Emission Factors for Road Transport Version 3 (HBEFA3) emission model, which estimates CO_2, NO_x, PM_x, and fuel consumption based on instantaneous vehicle speed and acceleration profiles. The GNN surrogate model was implemented in Python using PyTorch 2.1 and PyTorch Geometric 2.4. All experiments were executed on a workstation with an Intel Core i7-12700K CPU (12 cores, 3.6 GHz), 64 GB RAM, and an NVIDIA GeForce RTX 3080 GPU (10 GB VRAM), running Ubuntu 22.04 LTS.

4.2 | Test Networks

Three categories of test networks were used to evaluate GRAPH-TSC under diverse conditions:

Network A (Algiers downtown (42 intersections)): this network was modeled after the central business district of Algiers, Algeria, encompassing the Didouche Mourad Boulevard, Rue Larbi Ben M'hidi, and the surrounding grid of the Casbah periphery. Road geometry, lane configurations, and signal phase structures were extracted from OpenStreetMap data and validated through field observation. The network comprises 42 signalized intersections connected by 118 directed links. Intersection types include 2-phase (minor cross-streets), 3-phase (T-intersections), and 4-phase (major crossings) configurations. Traffic demand patterns were derived from traffic survey data provided by the Direction des Transports de la Wilaya d'Alger, covering morning peak (7:00–9:00), off-peak (10:00–16:00), and evening peak (17:00–19:00) periods. The total network demand ranges from approximately 8,500 vehicles per hour during off-peak to 19,200 vehicles per hour during the morning peak period.

Network B (Hanoi Old Quarter and Hoan Kiem district (67 intersections)): This network was modeled after the central area of Hanoi, Vietnam, encompassing the Old Quarter, the Hoan Kiem Lake vicinity, and portions of the Ba Dinh and Hai Ba Trung districts. The network features a highly irregular topology with a mix of narrow one-way streets, wide boulevards, and complex multi-leg intersections characteristic of Southeast Asian cities. A distinguishing feature of this network is the motorcycle-dominant mixed traffic composition, with an approximate modal split of 70% motorcycles, 25% passenger cars, and 5% buses. To accurately model this mixed traffic, SUMO's vehicle type parameters were calibrated using field data from the Hanoi Department of Transportation, including motorcycle-specific car-following parameters (shorter headways, more aggressive lane-changing behavior) and lane discipline characteristics. The network comprises 67 signalized intersections connected by 196 directed links, with total demand ranging from 12,000 to 28,500 PCU (passenger car units) per hour across demand periods.

Network C (Synthetic Grids (25, 64, 100 intersections)): three synthetic regular grid networks were constructed for controlled scalability analysis: a 5×5 grid (25 intersections), an 8×8 grid (64 intersections), and a 10×10 grid (100 intersections). All grid networks feature uniform geometry (4-phase intersections, two through lanes and one left-turn lane per approach, 300-meter block spacing, 50 km/h speed limit). Demand patterns were generated using a gravity model with random origin–destination weights to produce realistic but controlled traffic loading.

4.3 | Demand Scenarios

For each real network (A and B), five demand scenarios were defined to evaluate robustness:

Table 1. Demand scenarios used to evaluate the robustness of real networks A and B under typical, disrupted, event-driven, and evolving traffic conditions

Scenario	Description	Characteristics
S1: typical weekday	Standard AM peak, PM peak, off-peak	Baseline demand from traffic surveys
S2: weekend	Reduced commuter, increased recreational	~30% lower total demand, shifted temporal profile
S3: incident	Lane closure on major arterial	50% capacity reduction on one arterial link during peak
S4: special event	30% demand surge in one quadrant	Localized demand increase simulating stadium/event
S5: gradual shift	Evolving demand due to construction zone	Progressive demand redistribution over 2 hours

4.4 | Compared Methods

GRAPH-TSC was compared against eight baseline and state-of-the-art methods spanning fixed-time, adaptive, metaheuristic, and learning-based categories:

Webster Fixed-Time (Webster): optimal fixed-time signal timing computed using Webster's (1958) formulas based on measured traffic volumes at each intersection. Offsets optimized using SYNCHRO-style bandwidth maximization.

SCATS-like Adaptive (SCATS): an implementation of SCATS-like adaptive control, selecting from a library of pre-computed timing plans based on measured degree of saturation. Phase extension and early termination based on gap-out logic.

Standard ACO: conventional ACO for signal timing without graph-awareness, pheromone diffusion, or a GNN surrogate (using SUMO evaluation with a reduced number of iterations to maintain computational feasibility).

GA: standard GA with tournament selection, uniform crossover, and Gaussian mutation for signal timing parameters. Population size: 100, 500 generations, with SUMO-based fitness evaluation using abbreviated simulation runs.

PSO: standard PSO with 100 particles, inertia weight 0.7, cognitive and social coefficients 1.5, 500 iterations with SUMO-based fitness evaluation.

DQN Single-Intersection (DQN): independent DQN agents at each intersection, with state representation based on queue lengths and current phase, trained for 200 episodes on each network. Based on Li et al. [9] architecture.

CoLight-inspired Multi-Agent RL (CoLight): multi-agent RL with graph attention communication between intersection agents, based on the architecture of Wei et al. [12], trained for 500 episodes with parameter sharing across intersections of the same type.

MaxPressure: distributed control policy selecting the phase with maximum pressure at each intersection at each decision point Varaiya [34] No cycle constraint imposed; phases selected greedily every 5 seconds.

4.5 | Evaluation Metrics

The following metrics were used for performance evaluation, computed over the full simulation period (2 hours of simulated traffic) and averaged over 30 independent runs with different random seeds:

I. Average vehicle delay (seconds/vehicle): mean difference between actual and free-flow travel time per vehicle.

II. Network throughput (vehicles/hour): total number of vehicles completing their trips within the simulation period divided by simulation duration.

III. Average number of stops (stops/vehicle): mean number of complete stops per vehicle during its trip.

IV. Average travel time (seconds): mean end-to-end travel time across all vehicles.

V. CO_2 emissions (grams/km): average CO_2 emissions per vehicle-kilometer, computed using SUMO's HBEFA3 emission model.

VI. Average queue length (vehicles): mean queue length across all intersection approaches at peak demand.

VII. Re-optimization time (seconds): wall-clock time from initiation to completion of the optimization procedure.

4.6 | Algorithm Parameters

The GRAPH-TSC parameters used in all experiments are summarized in *Table 2*. Parameter sensitivity analysis indicated that the algorithm is robust to moderate perturbations of these values (see supplementary material).

Table 2. Parameters of the GRAPH-TSC algorithm used in all experiments.

Component	Parameter	Value
ACO	Number of ants (M)	50
Maximum iterations (I_{max})	200	
Pheromone exponent (α)	1.0	
Heuristic exponent (β)	2.0	
Evaporation rate (ϱ)	0.1	
Diffusion rate (ϱ_{diff})	0.3	
Discretization	Parameter bins (B)	20
Cycle length range	[60, 180] seconds	
GNN Surrogate	Retraining interval	30 min (simulated)

Table 2. Continued.

Component	Parameter	Value
Training samples	50,000	
Hierarchical	Subnetwork size	15–25 intersections
Overlap zone	3–5 intersections	
CUSUM	Threshold (h)	Calibrated per network
Local search	Max evaluations	100 per solution
Statistics	Independent runs	30

5 | Results

5.1 | Algiers Network Results

Table 2 presents the performance comparison of all methods on the Algiers Downtown network (Network A) during the AM peak demand scenario (S1). GRAPH-TSC achieves the lowest average vehicle delay (28.4 ± 1.2 seconds), representing a 31.2% reduction relative to Webster fixed-time (41.3 ± 0.8 seconds), an 18.1% reduction relative to SCATS-like adaptive control (34.7 ± 1.5 seconds), and an 8.9% reduction relative to CoLight multi-agent RL (31.2 ± 2.1 seconds). Network throughput under GRAPH-TSC reaches 14,850 ± 320 vehicles per hour, a 22.0% improvement over Webster and a 14.1% improvement over SCATS. CO_2 emissions are reduced by 19.2% relative to Webster and by 12.4% relative to SCATS.

Table 3. Performance comparison on the Algiers Downtown network (42 intersections, AM peak scenario). Values are mean ± standard deviation over 30 runs. Best values are shown in bold.

Method	Avg. Delay (s/veh)	Throughput (veh/h)	Stops (per veh)	CO_2 (g/km)	Queue (veh)	Optimal Time(s)
Webster	41.3 ± 0.8	12,170 ± 180	3.4 ± 0.2	218.5 ± 3.1	12.8 ± 0.9	N/A
SCATS-like	34.7 ± 1.5	13,020 ± 250	2.8 ± 0.3	201.3 ± 3.8	10.2 ± 1.1	N/A
Standard ACO	35.6 ± 1.8	12,780 ± 290	2.9 ± 0.3	205.7 ± 4.2	10.9 ± 1.3	2,640
GA	36.9 ± 1.6	12,540 ± 310	3.1 ± 0.3	210.4 ± 3.9	11.4 ± 1.2	3,120
PSO	37.4 ± 1.9	12,390 ± 280	3.2 ± 0.4	212.1 ± 4.5	11.7 ± 1.4	2,980
DQN	33.5 ± 2.3	13,280 ± 350	2.6 ± 0.4	196.8 ± 5.1	9.8 ± 1.5	N/A*
CoLight	31.2 ± 2.1	13,680 ± 380	2.4 ± 0.3	192.5 ± 4.7	9.1 ± 1.3	N/A*
MaxPressure	32.8 ± 1.7	13,410 ± 270	2.7 ± 0.3	198.2 ± 3.6	9.5 ± 1.0	N/A
GRAPH-TSC	28.4 ± 1.2	14,850 ± 320	2.1 ± 0.2	176.5 ± 2.8	7.3 ± 0.8	1.9

Note: DQN and CoLight require offline training: DQN ~4 hours, CoLight ~12 hours. Optimization time reflects online deployment only.

Notably, GRAPH-TSC outperforms both CoLight and MaxPressure despite not requiring any training phase. The standard metaheuristic methods (Standard ACO, GA, PSO) achieve moderate improvements over Webster but are outperformed by GRAPH-TSC due to their lack of graph awareness and inability to efficiently coordinate offsets across the network. Moreover, these methods require optimization times exceeding 40 minutes due to their reliance on SUMO-based fitness evaluation, precluding real-time application.

5.2 | Hanoi Network Results

Table 4 presents the performance comparison on the Hanoi network (Network B, 67 intersections) during the AM peak scenario. The larger network size and the challenges of motorcycle-dominant mixed traffic amplify the performance differences among methods. GRAPH-TSC achieves the lowest average delay (33.1 ± 1.5 seconds), a 37.3% reduction relative to Webster (52.8 ± 1.1 seconds), a 21.7% reduction relative to SCATS (42.3 ± 1.7 seconds), and a 12.4% reduction relative to CoLight (37.8 ± 2.4 seconds). The larger improvement margins on the Hanoi network relative to the Algiers network are attributable to the more complex topology (irregular grid, one-way streets) and the higher potential for coordination gains in a network where spatial dependencies are more pronounced.

Table 4. Performance comparison on the Hanoi network (67 intersections, AM peak scenario). Values are mean ±standard deviation over 30 runs.

Method	Average. Delay (s/veh)	Throughput (veh/h)	Stops (per veh)	CO_2 (g/km)	Queue (veh)	Optimal. Time (s)
Webster	52.8 ± 1.1	16,420 ± 240	4.6 ± 0.3	245.3 ± 3.5	16.5 ± 1.2	N/A
SCATS-like	42.3 ± 1.7	17,890 ± 310	3.7 ± 0.4	225.7 ± 4.1	13.1 ± 1.5	N/A
Standard ACO	44.1 ± 2.3	17,450 ± 350	3.9 ± 0.4	230.2 ± 4.8	13.8 ± 1.7	4,350
GA	45.8 ± 2.1	17,100 ± 380	4.1 ± 0.4	234.6 ± 5.2	14.5 ± 1.6	5,180
PSO	46.5 ± 2.4	16,930 ± 340	4.2 ± 0.5	237.1 ± 5.5	14.9 ± 1.8	4,780
DQN	40.7 ± 2.8	18,240 ± 420	3.5 ± 0.5	220.4 ± 5.8	12.4 ± 1.8	N/A*
CoLight	37.8 ± 2.4	18,870 ± 450	3.1 ± 0.4	214.2 ± 5.3	11.5 ± 1.6	N/A*
MaxPressure	39.2 ± 2.0	18,540 ± 330	3.4 ± 0.3	218.9 ± 4.2	12.0 ± 1.3	N/A
GRAPH-TSC	33.1 ± 1.5	20,080 ± 390	2.5 ± 0.3	199.8 ± 3.2	8.9 ± 1.0	2.7

Note: DQN training: ~7 hours; CoLight training: ~20 hours on the 67-intersection network.

The motorcycle-dominant traffic composition in the Hanoi network introduces additional complexity for all methods. Motorcycles exhibit smaller headways, more frequent lane changes, and less predictable trajectories than passenger cars. The graph-aware coordination in GRAPH-TSC proves particularly effective in this context, as the pheromone diffusion mechanism captures the flow-dependent spatial dependencies that are amplified by the high density of motorcycle traffic.

5.3 | Demand Scenario Robustness

Table 5 presents the average vehicle delay achieved by each method across all five demand scenarios on the Hanoi network. GRAPH-TSC demonstrates the best adaptability, with the smallest delay increases under non-recurrent conditions (incident and event scenarios). Under the incident scenario (S3), GRAPH-TSC's delay increases by only 18% relative to the typical weekday scenario, compared to +45% for Webster, +28% for SCATS, and +22% for CoLight. This robustness is attributed to the adaptive re-optimization trigger, which detects the traffic pattern shift caused by the incident and initiates a rapid re-optimization that redistributes signal timing to accommodate the disrupted traffic patterns.

Table 5. Average vehicle delay (s/veh) across demand scenarios on the Hanoi network (67 intersections). Best values per scenario in bold.

Method	S1: Typical	S2: Weekend	S3: Incident	S4: Event	S5: Shift
Webster	52.8	38.2	76.6	68.1	61.4
SCATS-like	42.3	30.5	54.1	51.8	47.6
CoLight	37.8	27.9	46.1	45.3	42.1
MaxPressure	39.2	28.4	49.8	47.5	43.7
GRAPH-TSC	33.1	24.8	39.1	39.7	36.5

Under the gradual shift scenario (S5), GRAPH-TSC's CUSUM trigger detects the evolving demand pattern and triggers 3–4 re-optimizations over the 2-hour simulation period, each requiring approximately 2.7 seconds. In contrast, Webster maintains its original fixed-time plan throughout, experiencing a cumulative delay increase of 16.3%. CoLight's RL policy adapts to a degree, but its performance degrades when the evolved demand pattern diverges significantly from the training distribution.

5.4 | Graph Neural Network Surrogate Accuracy

Table 6 quantifies the accuracy of the GNN surrogate model by comparing its predictions against full SUMO simulation outputs across both test networks. The surrogate achieves consistently low prediction errors, with MAE for average delay prediction of 2.3 seconds (7.1% relative error) on the Algiers network and 2.8 seconds (6.5% relative error) on the Hanoi network. These error levels are well within the stochastic variability inherent in microsimulation itself (typically 5–10% coefficient of variation across replications with different random seeds).

Table 6. GNN surrogate model accuracy: comparison of surrogate predictions vs, Full SUMO simulation across both networks.

Metric	Algiers (MAE)	Algiers (Relative Error)	Hanoi (MAE)	Hanoi (Relative Error)
Average delay (s/veh)	2.3	7.1%	2.8	6.5%
Queue length (vehicles)	1.1	9.3%	1.4	8.7%
Throughput (veh/h)	410	3.2%	520	2.8%
Surrogate evaluation time	~2 ms per plan	~2 ms per plan		
SUMO simulation time	~25 s per plan	~35 s per plan		
Speed-up factor	~12,500×	~17,500×		
Retraining time	~40 s (warm-start)	~50 s (warm-start)		

The speed-up factor of approximately 12,500–17,500× is the critical enabler of real-time optimization. Without the GNN surrogate, a single ACO run with 50 ants and 200 iterations would require 50 × 200 × 35 seconds ≈ 97 hours on the Hanoi network, clearly infeasible for real-time control. With the surrogate, the same computation requires less than 3 seconds. Periodic retraining every 30 minutes ensures that the surrogate remains calibrated as traffic conditions evolve, with retraining requiring only 40–50 seconds using warm-starting from the previous model weights.

5.5 | Scalability Analysis

Table 7 presents the scalability analysis on the synthetic grid networks, demonstrating GRAPH-TSC's performance and computational requirements as network size increases from 25 to 100 intersections.

Table 7. Scalability analysis on synthetic grid networks. Average delay (s/veh), throughput improvement over Webster (%), and re-optimization time (s).

Network Size	GRAPH-TSC Delay (s/veh)	Webster Delay (s/veh)	Delay Reduction (%)	Throughput Improvement (%)	Optimal. Time (s)	Decomposition
5 × 5 (25)	24.2 ± 0.9	33.5 ± 0.6	27.8	22.4	0.8	None
8 × 8 (64)	29.7 ± 1.3	40.1 ± 0.8	25.9	20.1	2.7	3 subnetworks
10 × 10 (100)	31.4 ± 1.6	42.3 ± 0.9	25.8	19.5	4.1	5 subnetworks

The results demonstrate that GRAPH-TSC scales gracefully with network size. The delay reduction percentage decreases only slightly from 27.8% on the 25-intersection grid to 25.8% on the 100-intersection grid, a quality degradation of less than 5%. The re-optimization time increases from 0.8 seconds to 4.1 seconds as the network grows fourfold, remaining well within real-time constraints. The hierarchical decomposition is activated for networks exceeding 50 intersections (the 8×8 and 10×10 grids), partitioning them into 3 and 5 subnetworks respectively, with boundary coordination converging within 3–4 hierarchical iterations.

5.6 | Ablation Study

Table 8 presents the results of an ablation study that systematically removes each component of GRAPH-TSC to quantify its individual contribution to overall performance. All experiments are conducted on the Hanoi network under the AM peak scenario.

Table 8. Ablation study on the Hanoi network (67 intersections, AM peak). Average delay (s/veh) and optimization time (s) for each ablated variant.

Variant	Average. Delay (s/veh)	Δ Delay (%)	Optimal. Time (s)
GRAPH-TSC (full)	33.1		2.7
Without GNN surrogate (SUMO eval.)	31.8	−3.9	2,720
Without pheromone diffusion	37.2	+12.4	2.5
Without hierarchical decomposition	35.9	+8.5	5.8
Without adaptive trigger (fixed 10-min)	34.5	+4.2	2.7
Without local search	34.8	+5.1	2.5

The ablation results reveal several insights. First, the variant without the GNN surrogate achieves slightly lower delay (31.8 s vs. 33.1 s) because it benefits from the higher fidelity of full SUMO evaluation. However, this comes at a prohibitive computational cost of 2,720 seconds (over 45 minutes), rendering it entirely impractical for real-time deployment. The GNN surrogate thus introduces a modest 3.9% quality trade-off in exchange for a 1,000× reduction in optimization time, a trade-off that is essential for real-time applicability. Second, pheromone diffusion is the most impactful algorithmic component, contributing a 12.4% delay reduction. The observed delay reduction confirms that the flow-dependent propagation of coordination information along the traffic graph is central to GRAPH-TSC's performance advantage. Third, hierarchical decomposition contributes an 8.5% delay reduction and also reduces optimization time from 5.8 seconds (monolithic) to 2.7 seconds by enabling parallelized subnetwork optimization. Fourth, the adaptive trigger and local search each contribute moderate improvements of 4.2% and 5.1% respectively.

5.7 | Green Wave Analysis

A key practical benefit of network-wide signal coordination is the creation of "green wave" sequences of coordinated signals along major arterials that allow platoons of vehicles to traverse multiple intersections without stopping. To quantify this coordination quality, we analyzed the green wave success rate on the five major arterials in the Algiers network, defined as the percentage of vehicles that traverse three or more consecutive intersections on these arterials without encountering a red signal.

GRAPH-TSC achieves a green wave success rate of 72% on the major arterials, compared to 45% for Webster fixed-time, 58% for SCATS-like adaptive, 63% for CoLight, and 66% for MaxPressure. The superior green wave performance of GRAPH-TSC is directly attributable to the pheromone diffusion mechanism, which explicitly propagates offset coordination information along high-flow corridors. In the ablated variant without pheromone diffusion, the green wave success rate drops to 51%, confirming the causal link between this mechanism and coordination quality.

On the Hanoi network, green wave analysis is more complex due to the irregular topology and the prevalence of one-way streets that create asymmetric flow patterns. Nevertheless, GRAPH-TSC achieves a 64% green wave success rate on the five highest-flow corridors in the Hanoi network, compared to 38% for Webster and 52% for SCATS-like adaptive. The larger improvement margin in Hanoi reflects the greater coordination potential in a network where the irregular topology creates more opportunities for intelligent offset optimization.

5.8 | Real-Time Performance

Table 9 presents the end-to-end response time of GRAPH-TSC from the initial detection of a traffic pattern shift to the deployment of a new optimized signal plan. The response time is decomposed into three phases: 1) change detection latency (time from the onset of the traffic pattern shift to the CUSUM threshold exceedance), 2) optimization time (execution of the GRAPH-TSC algorithm), and 3) plan deployment time (communication of the new plan to signal controllers via the central server).

Table 9. End-to-end response time decomposition for GRAPH-TSC on both test networks.

Phase	Algiers (42 Intervals)	Hanoi (67 Intervals)
Change detection (CUSUM lag)	10 s	12 s
Optimization (GRAPH-TSC)	1.9 s	2.7 s
Plan deployment (communication)	0.2 s	0.3 s
Total response time	12.1 s	15.0 s

The total response time of 12–15 seconds is well within the temporal resolution required for operational traffic signal management. Traffic signal plans are typically maintained for at least one full cycle (60–180 seconds) before transitioning to a new plan, and transitions themselves require phase-compatible handoffs that may span one additional cycle. Thus, a 15-second response time allows GRAPH-TSC to respond to traffic disruptions within a single cycle, a response speed comparable to or faster than commercial adaptive systems such as SCOOT (which typically adjusts parameters every 5 seconds incrementally) and SCATS (which selects plans every cycle).

6 | Discussion

The experimental results presented in Section 5 demonstrate that GRAPH-TSC achieves substantial and consistent performance improvements over established fixed-time, adaptive, metaheuristic, and learning-based traffic signal control methods across diverse network topologies, traffic compositions, and demand scenarios. In this section, we discuss the underlying factors contributing to these results, the practical implications for deployment, the limitations of the approach, and directions for future research.

Why does graph-aware ACO outperform standard metaheuristics? The performance advantage of GRAPH-TSC over standard ACO, GA, and PSO for signal optimization can be attributed to two complementary mechanisms. First, the graph-traversal-based solution construction ensures that when an ant selects the offset for an intersection, it does so in the context of already-determined offsets at adjacent intersections, naturally encoding the spatial dependency structure of the coordination problem. Standard metaheuristics treat each intersection's parameters as independent decision variables, ignoring the critical inter-intersection dependencies that determine coordination quality. Second, the pheromone diffusion mechanism provides an explicit information propagation channel along the traffic graph. When a timing plan achieves good performance at one intersection, the associated offset information diffuses to adjacent intersections weighted by traffic flow intensity. Over successive iterations, this diffusion process constructs "pheromone corridors" along major arterials that guide ants toward coordinated offset patterns, a behavior that emerges from the graph structure without requiring explicit green wave optimization formulations.

Comparison with RL-based approaches: GRAPH-TSC outperforms CoLight (the best-performing RL baseline) by 8.9% on the Algiers network and 12.4% on the Hanoi network, despite CoLight being trained for 500 episodes (12–20 hours). Several factors contribute to this performance gap. First, GRAPH-TSC optimizes explicit signal timing parameters (cycle, green split, offset), producing interpretable timing plans that can be reviewed, validated, and modified by traffic engineers. CoLight learns a black-box policy mapping observations to phase selections, which is less amenable to engineering validation. Second, GRAPH-TSC can be immediately deployed on a new network without any training phase; it only requires the GNN surrogate to be pre-trained on simulation data, which is a one-time cost that can be completed during the network modeling phase. CoLight requires network-specific training that must be repeated for each new deployment. Third, GRAPH-TSC's re-optimization produces a complete timing plan that is robust over the duration of its application (until the next re-optimization trigger), whereas RL-based methods make decisions at every time step, creating potential instability under communication delays or sensor failures. We note, however, that RL-based methods have the theoretical advantage of continuous adaptation and do not require explicit change detection triggers.

GNN surrogate as a key enabler: The GNN surrogate model is the linchpin that makes GRAPH-TSC viable for real-time application. Without it, the optimization would require over 45 minutes, well beyond any practical definition of real-time control. The modest accuracy trade-off (7.1% relative error) is well within acceptable limits for operational traffic management, where the inherent stochasticity of traffic flow introduces variability of similar magnitude. The periodic retraining mechanism ensures that the surrogate adapts to evolving traffic conditions, maintaining its accuracy over extended deployment periods. Future work could explore more sophisticated online learning strategies, such as Bayesian optimization of the retraining schedule or reinforcement of the surrogate in regions of the parameter space where prediction uncertainty is highest.

Handling motorcycle-dominant traffic: The Hanoi network experiments demonstrate GRAPH-TSC's effectiveness in motorcycle-dominant mixed traffic environments, which are characteristic of many developing cities in Southeast Asia, South Asia, and sub-Saharan Africa. Motorcycles present unique challenges for traffic signal optimization: their smaller physical size allows higher density in queue storage, their acceleration and deceleration characteristics differ substantially from passenger cars, and their lane discipline is significantly less structured. GRAPH-TSC's GNN surrogate captures these dynamics through its training on SUMO simulations calibrated with motorcycle-specific behavioral parameters. The larger improvement margins observed on the Hanoi network (relative to Algiers) suggest that graph-aware coordination provides proportionally greater benefits in complex, high-density traffic environments where spatial dependencies are amplified.

Limitations. Despite the demonstrated effectiveness of GRAPH-TSC, several limitations should be acknowledged. First, the GNN surrogate introduces an approximation error that, while modest on average, may be larger for extreme or out-of-distribution traffic conditions that were underrepresented in the training data. The periodic retraining mitigates this risk, but does not eliminate it entirely. Second, GRAPH-TSC assumes that signal phase structures (phase sequences, permitted/protected movements) are pre-specified and fixed during optimization. In practice, phase structures may need to be adapted in response to demand changes, a capability that is not currently supported. Third, pedestrian traffic is incorporated only through minimum green time constraints and is not explicitly optimized; dedicated pedestrian signal optimization (e.g., for diagonal crossings or leading pedestrian intervals) remains a topic for future investigation. Fourth, the current implementation assumes a centralized architecture in which a central server receives detector data from all intersections, executes the optimization, and communicates timing plans back to controllers. This architecture requires reliable, low-latency communication infrastructure, which may not be universally available, particularly in developing cities. Fifth, the current formulation does not incorporate transit signal priority or emergency vehicle preemption, which are important operational requirements in many urban networks.

Practical deployment considerations. Deploying GRAPH-TSC in a real-world traffic management center requires several infrastructure prerequisites: 1) loop or video detectors at all signalized intersections to provide real-time demand data), 2) a central traffic management server with GPU capability (the GNN surrogate requires GPU for efficient inference, though CPU inference is feasible at approximately 10× slower speed), 3) a reliable communication network between the central server and signal controllers (standard National Transportation Communications for Intelligent Transportation System Protocol (NTCIP) protocols over fiber or cellular networks are sufficient), and 4) a fail-safe mechanism that reverts to pre-computed fixed-time plans in case of communication failure or optimization anomalies. The total hardware cost for the central server component is estimated at approximately United States Dollar (USD) 5,000–8,000, which is modest relative to the cost of deploying commercial adaptive systems such as SCOOT or SCATS. In case of failure, GRAPH-TSC degrades gracefully: the most recently deployed timing plan continues to operate as a fixed-time plan until communication is restored or a manual override is initiated.

Future directions. Several promising extensions of GRAPH-TSC merit investigation. First, a decentralized multi-swarm variant could distribute the ACO computation across edge devices at intersection controllers, eliminating the central server requirement and improving robustness to communication failures. Each subnetwork could maintain its own ant colony, with boundary coordination achieved through lightweight pheromone exchange messages between adjacent controllers. Second, integration with Connected and Autonomous Vehicle (CAV) data could enhance the GNN surrogate's state estimation accuracy and provide advance demand information through trajectory data from connected vehicles. Third, multi-modal transit priority could be incorporated by extending the objective function to include transit delay penalties and adding transit priority constraints to the optimization formulation. Fourth, Vehicle-To-Infrastructure (V2I) communication could enable demand prediction at longer horizons, allowing GRAPH-TSC to optimize proactively rather than reactively. Fifth, extension to networks containing unsignalized intersections (roundabouts, yield-controlled) through coupled simulation of signalized and unsignalized intersection models would improve the accuracy of network-wide performance prediction. Finally, multi-objective optimization using Pareto-based ACO variants inspired by NSGA-II [37] could explicitly trade off delay, emissions, equity, and transit priority, providing decision-makers with a Pareto front of non-dominated solutions.

7 | Conclusion

This paper has presented GRAPH-TSC, a novel graph-based metaheuristic optimization algorithm for real-time urban GRAPH-TSC. GRAPH-TSC models the traffic network as a directed graph and employs a graph-aware variant that traverses the traffic graph during solution construction, naturally capturing the spatial dependencies among intersections that are critical for effective signal coordination. The algorithm introduces four key innovations: a pheromone diffusion mechanism that propagates coordination information along high-flow corridors to facilitate green wave formation; a GAT-based surrogate model that enables real-time fitness evaluation by replacing computationally expensive microsimulation; a hierarchical network decomposition strategy that partitions large networks into overlapping subnetworks for scalable optimization; and an adaptive re-optimization trigger based on CUSUM change detection that initiates re-optimization only when traffic pattern shifts are detected.

Extensive experiments on SUMO-simulated networks modeled after central Algiers (42 intersections) and central Hanoi (67 intersections), as well as synthetic grid networks of up to 100 intersections, demonstrate that GRAPH-TSC achieves 18–27% reductions in average vehicle delay, 14–22% improvements in network throughput, and 11–19% reductions in CO_2 emissions compared to Webster's fixed-time, SCATS-like adaptive, standard ACO, GA, and DQN-based control methods. GRAPH-TSC also outperforms state-of-the-art multi-agent RL (CoLight) by 9–12% in delay reduction, without requiring any offline training phase. The algorithm completes re-optimization within 2.7 seconds for the 67-intersection Hanoi network, with a total end-to-end response time of 15 seconds from pattern shift detection to new plan deployment. The hierarchical decomposition enables graceful scaling to 100-intersection networks with less than 5% quality degradation.

GRAPH-TSC makes a distinctive contribution to the field of intelligent transportation systems by demonstrating that graph-aware metaheuristic optimization, when augmented with GNN-based surrogate models, can achieve performance competitive with or superior to deep RL methods while maintaining the interpretability, immediate deployability, and computational efficiency required for practical traffic management in smart cities. The algorithm is particularly well-suited for deployment in developing cities where traffic patterns are complex and dynamic, communication infrastructure may be limited, and the absence of extensive training data precludes the direct application of learning-based methods. The open-source release of the simulation framework and algorithm implementation is intended to facilitate reproducibility and further development by the research community.

Acknowledgments

This research was supported by the Algerian Ministry of Higher Education and Scientific Research under PRFU Grant A25N01UN160120230001, and by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant 102.01-2024.15. The authors thank the Direction des Transports de la Wilaya d'Alger and the Hanoi Department of Transportation for providing traffic survey data used in this study. The authors also thank the anonymous reviewers for their constructive comments and suggestions.

Author Contributions

H. B.: conceptualization, methodology, software, writing original draft, supervision, project administration. N. M. D.: data curation, software, validation, formal analysis, writing review & editing, visualization.

Funding

Not applicable.

Data Availability

The SUMO simulation network files, algorithm source code, and GNN surrogate model training scripts are publicly available at github.com/boussouf-lab/GRAPH-TSC. Raw traffic demand data for the Algiers network are available upon request from the Direction des transports de la Wilaya d'Alger. Traffic demand data for the Hanoi network are available upon request from the Hanoi Department of Transportation. Synthetic grid network generation scripts are included in the repository.

Competing Interests

The authors declare no competing interests.

Consent for Publication

The author confirms consent for the publication of this work.

Ethics Approval and Consent to Participate

This article does not contain any studies with human participants performed by the author.

Reference

[1] Schrank, D., Eisele, B., & Lomax, T. (2021). *2021 urban mobility report.* https://static.tti.tamu.edu/tti.tamu.edu/documents/umr/archive/mobility-report-2021.pdf

[2] Cervero, R. (2013). Linking urban transport and land use in developing countries. *Journal of transport and land use*, *6*(1), 7–24. https://www.jstor.org/stable/26202644

[3] Nguyen, H. D. (2016). Saturation flow rate analysis at signalized intersections for mixed traffic conditions in motorcycle dependent cities. *Transportation research procedia*, *15*, 694–708. https://doi.org/10.1016/j.trpro.2016.06.058

[4] Federal Highway Administration (FHWA). (2008). *Traffic signal timing manual*. https://ops.fhwa.dot.gov/publications/fhwahop08024/fhwa_hop_08_024.pdf

[5] Hunt, P. B., Robertson, D. I., Bretherton, R. D., & Royle, M. C. (1982). The SCOOT on-line traffic signal optimisation technique. *Traffic engineering & control*, *23*(4), 190-192. http://worldcat.org/issn/00410683

[6] Lowrie, P. R. (1990). *Scats, sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic*. https://trid.trb.org/View/488852

[7] Stevanovic, A. (2010). *Adaptive traffic control systems: Domestic and foreign state of practice*. Transportation Research Board (TRB), National Academies. https://nap.nationalacademies.org/catalog/14364

[8] Head, K. L., Mirchandani, P. B., & Sheppard, D. (1992). Hierarchical framework for real-time traffic control. *Transportation research record, 1360*, 82–88. https://onlinepubs.trb.org/Onlinepubs/trr/1992/1360/1360-014.pdf

[9] Li, L., Lv, Y., & Wang, F. Y. (2016). Traffic signal timing via deep reinforcement learning. *IEEE/CAA journal of automatica sinica*, *3*(3), 247–254. https://doi.org/10.1109/JAS.2016.7508798

[10] El-Tantawy, S., Abdulhai, B., & Abdelgawad, H. (2013). Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown Toronto. *IEEE transactions on intelligent transportation systems*, *14*(3), 1140–1150. https://doi.org/10.1109/TITS.2013.2255286

[11] Chu, T., Wang, J., Codecà, L., & Li, Z. (2019). Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE transactions on intelligent transportation systems*, *21*(3), 1086–1095. https://doi.org/10.1109/TITS.2019.2901791

[12] Wei, H., Xu, N., Zhang, H., Zheng, G., Zang, X., Chen, C., ... & Li, Z. (2019). Colight: Learning network-level cooperation for traffic signal control. *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1913-1922). Association for Computing Machinery (ACM). https://doi.org/10.1145/3357384.3357902

[13] Wei, H., Chen, C., Zheng, G., Wu, K., Gayah, V., Xu, K., & Li, Z. (2019). Presslight: Learning max pressure control to coordinate traffic signals in arterial network. *Proceedings of the 25th ACM SIGKDD international conference on*

knowledge discovery & data mining (pp. 1290-1298). Association for Computing Machinery (ACM). https://doi.org/10.1145/3292500.3330949

[14] Oroojlooy, A., Nazari, M., Hajinezhad, D., & Silva, J. (2020). Attendlight: Universal attention-based reinforcement learning model for traffic signal control. *Advances in neural information processing systems*, *33*, 4079–4090. https://proceedings.neurips.cc/paper/2020/file/29e48b79ae6fc68e9b6480b677453586-Paper.pdf

[15] Lu, J., Ruan, J., Jiang, H., Li, Z., Mao, H., & Zhao, R. (2023). *Dualight: Enhancing traffic signal control by leveraging scenario-specific and scenario-shared knowledge*. https://doi.org/10.48550/arXiv.2312.14532

[16] Noaeen, M., Naik, A., Goodman, L., Crebo, J., Abrar, T., Abad, Z. S. H., ... & Far, B. (2022). Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert systems with applications*, *199*, 116830. https://doi.org/10.1016/j.eswa.2022.116830

[17] Ceylan, H., & Bell, M. G. H. (2004). Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing. *Transportation research part b: Methodological*, *38*(4), 329–342. https://doi.org/10.1016/S0191-2615(03)00015-8

[18] Park, B., Messer, C. J., & Urbanik, T. (2000). Enhanced genetic algorithm for signal-timing optimization of oversaturated intersections. *Transportation research record*, *1727*(1), 32-41. https://doi.org/10.3141/1727-05

[19] Garcia-Nieto, J., Olivera, A. C., & Alba, E. (2013). Optimal cycle program of traffic lights with particle swarm optimization. *IEEE transactions on evolutionary computation*, *17*(6), 823–839. https://doi.org/10.1109/TEVC.2013.2260755

[20] Dorigo, M. (2007). Ant colony optimization. *Scholarpedia*, *2*(3), 1461. http://dx.doi.org/10.4249/scholarpedia.1461

[21] D'Acierno, L., Gallo, M., & Montella, B. (2012). An ant colony optimisation algorithm for solving the asymmetric traffic assignment problem. *European journal of operational research*, *217*(2), 459–469. https://doi.org/10.1016/j.ejor.2011.09.035

[22] Osorio, C., & Bierlaire, M. (2013). A simulation-based optimization framework for urban transportation problems. *Operations research*, *61*(6), 1333–1345. https://doi.org/10.1287/opre.2013.1226

[23] Kipf, T. N., & Welling, M. (2016). *Semi-supervised classification with graph convolutional networks*. https://doi.org/10.48550/arXiv.1609.02907

[24] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). *Graph attention networks*. https://doi.org/10.48550/arXiv.1710.10903

[25] Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). *Diffusion convolutional recurrent neural network: Data-driven traffic forecasting*. https://doi.org/10.48550/arXiv.1707.01926

[26] Yu, B., Yin, H., & Zhu, Z. (2017). *Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting*. https://doi.org/10.48550/arXiv.1709.04875

[27] Manual, H. C. (2000). Highway capacity manual. *Washington, Dc*, *2*(1), 1. https://onlinepubs.trb.org/Onlinepubs/trnews/trnews193-004.pdf

[28] Little, J. D. C. (1966). The synchronization of traffic signals by mixed-integer linear programming. *Operations research*, *14*(4), 568–594. https://doi.org/10.1287/opre.14.4.568

[29] Gartner, N. H., Assman, S. F., Lasaga, F., & Hou, D. L. (1991). A multi-band approach to arterial traffic signal optimization. *Transportation research part b: Methodological*, *25*(1), 55–74. https://doi.org/10.1016/0191-2615(91)90013-9

[30] Zhang, L., Song, Z., Tang, X., & Wang, D. (2016). Signal coordination models for long arterials and grid networks. *Transportation research part c: Emerging technologies*, *71*, 215–230. https://doi.org/10.1016/j.trc.2016.07.015

[31] TS, N. S. P. (2003). *NEMA standards publication TS 2-2003: Traffic controller assemblies with NTCIP requirements*. National Electrical Manufacturers Association (NEMA). https://www.nema.org/docs/default-source/standards-document-library/nema-ts-2-amendment-3-watermarked.pdf

[32] Aboudolas, K., Papageorgiou, M., & Kosmatopoulos, E. (2009). Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation research part c: Emerging technologies*, *17*(2), 163–174. https://doi.org/10.1016/j.trc.2008.10.002

[33] Lin, S., De Schutter, B., Xi, Y., & Hellendoorn, H. (2012). Efficient network-wide model-based predictive control for urban traffic networks. *Transportation research part c*, *24*, 122–140. https://doi.org/10.1016/j.trc.2012.02.003

[34] Varaiya, P. (2013). Max pressure control of a network of signalized intersections. *Transportation research part c: Emerging technologies*, *36*, 177–195. https://doi.org/10.1016/j.trc.2013.08.014

[35] Zheng, G., Xiong, Y., Zang, X., Feng, J., Wei, H., Zhang, H., ... & Li, Z. (2019). Learning phase competition for traffic signal control. *Proceedings of the 28th ACM international conference on information and knowledge*

management (pp. 1963-1972). Association for Computing Machinery (ACM). https://doi.org/10.1145/3357384.3357900

[36] Foy, M. D., Benekohal, R. F., & Goldberg, D. E. (1992). Signal timing determination using genetic algorithms. *Transportation research record*, (1365), 108. http://onlinepubs.trb.org/Onlinepubs/trr/1992/1365/1365-011.pdf

[37] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2), 182–197. https://doi.org/10.1109/4235.996017

[38] Jia, H., Lin, Y., Luo, Q., Li, Y., & Miao, H. (2019). Multi-objective optimization of urban road intersection signal timing based on particle swarm optimization algorithm. *Advances in mechanical engineering*, *11*(4), 1687814019842498. https://doi.org/10.1177/1687814019842498

[39] Khamis, M. A., & Gomaa, W. (2014). Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Engineering applications of artificial intelligence*, *29*, 134–151. https://doi.org/10.1016/j.engappai.2014.01.007

[40] Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). *Graph wavenet for deep spatial-temporal graph modeling*. https://doi.org/10.48550/arXiv.1906.00121

[41] Zheng, C., Fan, X., Wang, C., & Qi, J. (2020). Gman: A graph multi-attention network for traffic prediction. *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 01, pp. 1234-1241). Association for the Advancement of Artificial Intelligence (AAAI). https://doi.org/10.1609/aaai.v34i01.5477

[42] Liang, X., Du, X., Wang, G., & Han, Z. (2019). A deep reinforcement learning network for traffic light cycle control. *IEEE transactions on vehicular technology*, *68*(2), 1243–1253. https://doi.org/10.1109/TVT.2018.2890726

[43] Narayanan, S., Makarov, N., & Antoniou, C. (2024). Graph neural networks as strategic transport modelling alternative - A proof of concept for a surrogate. *IET intelligent transport systems*, *18*(11), 2059–2077. https://doi.org/10.1049/itr2.12551

[44] Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y. P., Hilbrich, R., ... & Wießner, E. (2018). Microscopic traffic simulation using sumo. *2018 21st international conference on intelligent transportation systems (ITSC)* (pp. 2575-2582). IEEE. https://doi.org/10.1109/ITSC.2018.8569938

[45] Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. https://doi.org/10.48550/arXiv.1412.6980

[46] Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The computer journal*, *7*(4), 308–313. https://doi.org/10.1093/comjnl/7.4.308