Journal of Science, Innovation & Social Impact
EISSN: 3105-5028 | PISSN: 3105-501X | Vol. 2, No. 1 (2026)

Article

A Comparative Study of Reinforcement Learning and Metaheuristic Algorithms for Real-Time Last-Mile Delivery Scheduling

Jialu Wang [1,*]

[1] Business Administration, Fordham University, NY, USA
* Correspondence: Jialu Wang, Business Administration, Fordham University, NY, USA

Abstract: The exponential growth of e-commerce has intensified the need for efficient real-time delivery-scheduling algorithms that can handle dynamic urban logistics environments. This study presents a comprehensive comparative analysis of reinforcement learning and metaheuristic algorithms for last-mile delivery optimization under time-sensitive constraints. We implement and evaluate Adaptive Large Neighborhood Search (ALNS) and Deep Q-Network (DQN) approaches on benchmark instances representing realistic urban delivery scenarios with 50-200 customer nodes. The experimental results demonstrate that ALNS achieves superior solution quality with an average optimality gap of 3.2% under an approximately 5-second (≤6 s) operational time budget, while DQN exhibits better runtime scalability for networks exceeding 150 customers (in the stress-test setting). The sensitivity analysis reveals critical trade-offs between computational efficiency and solution robustness under varying traffic conditions. This research provides empirical guidelines for logistics practitioners to select appropriate algorithms based on operational constraints, thereby contributing to sustainable urban transportation systems.

Keywords: last-mile delivery scheduling, Reinforcement learning, Metaheuristic algorithms, Dynamic vehicle routing, Real-time optimization

Received: 14 January 2026
Revised: 04 March 2026
Accepted: 14 March 2026
Published: 18 March 2026

1. Introduction

1.1. Background and Motivation for Real-Time Delivery Optimization

1.1.1. Growth of E-Commerce and Increasing Demand for Efficient Last-Mile Delivery

The rapid expansion of digital commerce has fundamentally reshaped consumer expectations for delivery services. Recent industry statistics indicate that same-day and next-day delivery options now account for more than 60% of urban e-commerce transactions, placing significant operational pressure on logistics providers to enhance efficiency. Last-mile delivery represents roughly 53% of total shipping costs and is a major contributor to urban traffic congestion and carbon emissions. In metropolitan areas, a typical delivery vehicle completes between 80 and 120 stops per day, with route optimization directly influencing profitability, which generally ranges from 8% to 12% for leading logistics operators.

1.1.2. Challenges in Dynamic Urban Logistics Environments

Urban delivery operations face multiple complexities that distinguish them from traditional routing problems. Traffic congestion exhibits high temporal and spatial variability, with average speeds fluctuating between 15 and 35 kilometers per hour

depending on time of day and road characteristics. Vehicle capacity must accommodate not only volume and weight constraints but also special handling requirements such as temperature-controlled transport for sensitive goods. The stochastic nature of delivery further complicates planning, as actual service times at customer locations often deviate from estimates, with standard deviations reaching approximately 40% of the mean. Additionally, about 8% of orders experience last-minute cancellations or address changes, necessitating rapid and flexible route adjustments.

1.2. Research Gap and Problem Statement

1.2.1. Limitations of Existing Algorithm Comparison Studies

Previous evaluations of delivery scheduling algorithms exhibit methodological limitations that restrict their practical relevance. Many studies assess algorithms exclusively using static benchmark instances that fail to capture the dynamic complexities of real-world operations [1]. Furthermore, most investigations employ arbitrary parameter settings without systematic tuning procedures, making it challenging to determine whether observed performance differences result from intrinsic algorithmic strengths or merely configuration artifacts.

1.2.2. Need for Systematic Performance Evaluation under Real-Time Constraints

The effective deployment of delivery scheduling algorithms necessitates rigorous evaluation under time-sensitive conditions that closely mirror operational reality. Dispatchers typically have only 1 to 5 seconds to make routing decisions when new orders are received or disruptions occur, yet most studies allow unlimited computation time. Real-time evaluation should consider multiple performance dimensions beyond traditional optimality-gap metrics, including computational stability under diverse problem characteristics, memory usage patterns, and sensitivity to parameter variations.

1.3. Research Objectives and Contributions

1.3.1. Defining the Scope of Algorithm Comparison

This study focuses on a structured comparison between two representative algorithmic paradigms: Adaptive Large Neighborhood Search (ALNS) as an example of metaheuristic methods, and Deep Q-Network (DQN) as a reinforcement learning approach. The research problem encompasses dynamic vehicle routing with time windows, integrating realistic operational constraints such as vehicle capacity, driver shift durations, and time-dependent travel speeds. Primary experiments cover 50 to 200 customer locations, reflecting the typical daily workload of small to medium-sized delivery operations. An additional scalability test extends to 250-300 customers for runtime and resource profiling.

1.3.2. Overview of Experimental Methodology and Expected Contributions

Both algorithms are implemented using consistent programming frameworks and computational platforms to ensure a fair comparison. Systematic parameter tuning is applied on shared validation datasets, with final evaluation conducted on independent test sets. The study aims to provide empirical evidence characterizing performance trade-offs between metaheuristic and reinforcement learning approaches under realistic operational conditions, offering quantitative guidance for algorithm selection based on problem size, computation time constraints, and operational priorities.

2. Literature Review

2.1. Metaheuristic Algorithms for Vehicle Routing Problems

2.1.1. Adaptive Large Neighborhood Search and Its Variants

Adaptive Large Neighborhood Search (ALNS) has emerged as one of the most effective metaheuristic frameworks for vehicle routing optimization. Its core mechanism iteratively destroys and repairs partial solutions using a portfolio of removal and insertion

operators, with selection probabilities adjusted according to historical performance. Adaptive variants incorporate dynamic parameter adjustment mechanisms to achieve competitive results on dynamic vehicle routing instances [2]. Recent developments emphasize sophisticated operator design and adaptive weight update schemes, with destroy operators typically removing between 10% and 40% of customer assignments.

2.1.2. Variable Neighborhood Search Approaches

Variable Neighborhood Search (VNS) systematically explores multiple neighborhood structures to escape local optima and identify high-quality solutions. Hybrid approaches combine VNS with learning-based guidance to enhance performance in time-constrained environments. Such methods have demonstrated solution quality improvements of 15%-25% relative to traditional VNS implementations, while reducing computational time by roughly 40% [3].

2.1.3. Genetic Algorithms and Hybrid Methods

Genetic algorithms maintain populations of candidate solutions that evolve through selection, crossover, and mutation operations. Hybrid strategies integrate genetic algorithms with local search operators, applying local improvement procedures to offspring solutions before fitness evaluation. These methods have proven effective in maintaining solution diversity and mitigating premature convergence in complex delivery scenarios [4].

2.2. Reinforcement Learning Approaches for Delivery Scheduling

2.2.1. Deep Q-Network-Based Dispatching Methods

Deep Q-Networks (DQN) enable effective function approximation in high-dimensional state spaces through deep neural network architectures. Recent implementations employ multiple hidden layers to process state features, including current route configurations, pending customer requests, and traffic condition indicators. DQN-based approaches have been increasingly adopted for dynamic dispatching problems, demonstrating scalability and robustness under stochastic operational conditions [5].

2.2.2. Policy Gradient Methods for Routing Optimization

Policy gradient methods optimize parameterized policies directly through gradient ascent on the expected cumulative reward. Advanced architectures, such as graph neural network-based optimization, allow policy gradient methods to handle the combinatorial complexity of large-scale order assignment efficiently. These approaches support real-time decision-making while maintaining solution quality within approximately 5% of offline optimization benchmarks [6].

2.3. Existing Comparative Studies and Their Limitations

2.3.1. Summary of Benchmark Datasets and Evaluation Metrics

The vehicle routing research community has developed standardized benchmark suites to facilitate algorithm comparison. Solomon instances remain widely used, despite being designed for static routing scenarios. Adaptive large neighborhood search variants and other metaheuristics have been evaluated on extended benchmark suites, achieving average gaps below 2% from best-known solutions [7]. Evaluation metrics typically focus on solution quality measured by total travel distance, route duration, or vehicle count relative to best-known solutions on static benchmarks, or relative to a common reference solver for dynamic instances.

2.3.2. Identified Gaps in Current Comparison Methodologies

Current algorithm comparison studies exhibit methodological limitations that reduce their relevance for practical deployment. While machine learning-enhanced combinatorial

optimization has been shown to generate near-optimal solutions within strict real-time constraints, many comparative studies fail to fully account for both training and inference phases. Efficient approaches for amortizing training costs across multiple problem instances and comprehensive evaluation frameworks remain important areas for future research [8,9].

3. Methodology

3.1. Problem formulation for dynamic delivery scheduling

3.1.1. Mathematical definition of the dynamic VRPTW

The dynamic vehicle routing problem with time windows extends the classical VRPTW formulation to incorporate temporal customer revelation and time-dependent parameters. The problem is formulated over the planning horizon $H = [0, T]$, representing the operational window during which delivery services occur. A fleet of m homogeneous vehicles with capacity Q operates from a central depot denoted as node 0. Customer requests arrive dynamically throughout the planning period, with customer i becoming known at revelation time r_i. Each customer i requires service quantity $q_i \leq Q$ and specifies a time window $[e_i, l_i]$ during which service must commence.

The transportation network represents a complete directed graph $G = (V, A)$ where $V = \{0, 1, ..., n\}$ denotes the node set. Arc (i,j) possesses a time-dependent travel time $t_{ij}(\tau)$ representing the duration required to traverse from node i to node j when departing at time τ. The objective minimizes total operational costs, comprising distance-based vehicle operating expenses and time-based driver labor costs, subject to capacity constraints, time-window restrictions, and route-continuity conditions.

3.1.2. Objective function design balancing cost and service quality

The objective function integrates multiple stakeholder priorities, including operational efficiency, customer satisfaction, and service quality maintenance. The primary cost component consists of distance-based vehicle operating expenses, covering fuel consumption, maintenance, and depreciation, averaging \$0.85 to \$1.20 per kilometer. The temporal cost component accounts for driver labor expenses, with hourly rates ranging from \$18 to \$28. Service quality is incorporated through soft constraints that penalize late arrivals and excessively long routes. Similar multi-component objective functions have been employed in previous studies, reporting optimal weight ratios of approximately 0.50:0.35:0.15 for distance, time, and vehicle components [10].

3.2. Algorithm implementation and parameter configuration

3.2.1. ALNS implementation with adaptive operator selection

The Adaptive Large Neighborhood Search implementation employs a diverse operator portfolio encompassing seven removal operators and five insertion operators. The removal operators include random removal, worst removal, removal of customers whose removal yields the maximum cost reduction, Shaw removal targeting customers with similar characteristics, route removal, time-oriented removal, and cluster removal. Each operator maintains a weight parameter that is updated adaptively based on performance history via exponential smoothing with $\lambda = 0.95$.

The repair phase employs regret-based insertion, which evaluates the cost increase associated with inserting each unrouted customer. The 3-regret criterion provides an effective balance between computational overhead and solution quality. The acceptance criterion follows simulated annealing with an exponential cooling schedule, an initial temperature set to 10% of the initial solution cost, and a cooling rate of 0.98 applied every 50 iterations.

Table 1 presents the complete ALNS parameter configuration employed in this study.

Table 1. ALNS Algorithm Parameter Configuration.

Parameter Category	Parameter Name	Value	Rationale
Operator Selection	Number of Removal Operators	6	Diversity in solution destruction
Operator Selection	Number of Insertion Operators	5	Multiple construction strategies
Operator Selection	Weight Smoothing Factor (λ)	0.95	Balance history and recent performance
Adaptive Weights	New Best Solution Score	33	Strong reward for global improvements
Adaptive Weights	Improvement Score	9	Moderate reward for local gains
Adaptive Weights	Accepted Solution Score	3	Minimal reward for diversification
Removal Parameters	Destroy Percentage Range	15-40%	Adaptive neighborhood sizing
Insertion Parameters	Regret Level (k)	3	Computational efficiency trade-off
Acceptance Criterion	Initial Temperature	0.10 C_initial	Problem-scaled starting temperature
Acceptance Criterion	Cooling Rate	0.98	Gradual temperature reduction
Termination	Maximum Iterations	5000 (primary experiments)	Computational budget allocation

3.2.2. DQN-based dispatching with state representation design

The Deep Q-Network architecture processes state representations that encode relevant problem features through a multi-layer neural network with three hidden layers containing 256, 128, and 64 neurons, respectively, using ReLU activation. Deep Q-learning for same-day delivery has been shown to achieve competitive performance with similar architectural configurations [11]. The state representation includes spatial features, such as normalized customer coordinates and fixed-length distance summaries (e.g., depot-distance statistics and k-nearest-neighbor distance features), temporal features encoding remaining time-window slack, and capacity features tracking remaining vehicle capacity. At each decision step, the action selects the next customer for the active vehicle from the feasible candidate set, with infeasible actions masked to ensure VRPTW feasibility.

The training process employs experience replay with a buffer capacity of 100,000 state-action-reward-next_state tuples. The target network is updated every 1,000 training steps via a soft update with parameter $\tau = 0.005$. The discount factor $\gamma = 0.95$ balances immediate and future rewards. Previous studies have shown that carefully calibrated reward shaping can significantly accelerate convergence while maintaining solution quality [12].

Table 2 summarizes the optimized DQN hyperparameters identified through the tuning process.

Table 2. Optimized DQN Hyperparameters.

Hyperparameter Category	Parameter	Optimal Value	Search Range
Network Architecture	Hidden Layer Sizes	256, 128, 64	[64-512] per layer
Network Architecture	Activation Function	ReLU	{ReLU, tanh, ELU}
Learning Process	Learning Rate	0.0001	[0.00001, 0.001]
Learning Process	Batch Size	64	[16, 128]
Learning Process	Discount Factor (γ)	0.95	[0.90, 0.99]
Experience Replay	Buffer Capacity	100,000	[10,000, 500,000]
Target Network	Update Frequency	1,000 steps	[500, 5,000]
Target Network	Soft Update (τ)	0.005	[0.001, 0.01]
Exploration	ε-decay Rate	0.995	[0.99, 0.999]

3.2.3. Parameter tuning strategies for fair comparison

Fair algorithm comparison requires systematic parameter optimization using consistent methodologies and validation datasets. The parameter tuning process employs a two-stage approach: coarse grid search followed by fine-tuning via local optimization. The validation dataset comprises 30 problem instances with diverse characteristics, including customer counts ranging from 50 to 150, time-window densities from loose to tight, and geographical distributions including random, clustered, and mixed patterns. Previous studies have shown that systematic parameter tuning can improve algorithm performance by 18%-35% relative to default configurations, highlighting the importance of rigorous calibration procedures [13].

3.3. Experimental design and evaluation framework

3.3.1. Benchmark instance generation based on urban delivery scenarios

The experimental evaluation employs a comprehensive benchmark suite designed to represent realistic urban delivery operations across multiple operational scales and geographical contexts. The instance-generation process creates problem sets that reflect three metropolitan archetypes: dense urban cores with short inter-customer distances, suburban regions with moderate spatial dispersion, and mixed environments. Customer counts vary across {50, 75, 100, 125, 150, 175, 200}, representing typical daily workloads. Time window generation samples window widths from {30, 60, 90, 120} minutes with probabilities {0.2, 0.4, 0.3, 0.1}.

Table 3 provides detailed specifications for the complete benchmark instance suite.

Table 3. Benchmark Instance Specifications.

Instance Class	Customer Count	Geography Type	Time Window Density	Dynamic Degree	Replications
Small-Dense	50	Dense Urban	60% tight	70% dynamic	30
Small-Suburban	50	Suburban	40% tight	60% dynamic	30
Medium-Dense	100	Dense Urban	70% tight	75% dynamic	30
Medium-Suburban	100	Suburban	50% tight	65% dynamic	30
Medium-Mixed	100	Mixed	55% tight	70% dynamic	30
Large-Dense	150	Dense Urban	65% tight	80% dynamic	30
Large-Suburban	150	Suburban	45% tight	70% dynamic	30
XLarge-Dense	200	Dense Urban	60% tight	85% dynamic	30

3.3.2. Performance metrics definition, including solution quality and computational time

The evaluation framework employs multidimensional performance metrics that capture both solution quality and computational efficiency. Solution quality assessment computes the relative optimality gap: Gap = 100·(Algorithm_Cost - Reference_Cost)/Reference_Cost. Reference_Cost is defined as the best objective value obtained for the same instance by running ALNS with an extended time budget (60 seconds) and multiple random restarts, providing a consistent reference for gap computation in dynamic settings.

Computational efficiency metrics measure elapsed wall-clock time from problem initiation to solution return. Anytime performance tracking records solution quality at regular time intervals (1, 2, 3, 5 seconds). Robustness assessment evaluates algorithm stability across stochastic problem variations through the coefficient of variation $CV = \sigma/\mu$ across 30 replications.

3.3.3. Statistical analysis methods for result validation

The statistical validation employs rigorous hypothesis testing procedures to establish significant performance differences between algorithms. Paired t-tests compare mean solution costs across matched instance replications at a significance level of $\alpha = 0.05$. Analysis of variance (ANOVA) examines performance variations across instance classes, with post-hoc Tukey HSD tests identifying specific instance classes exhibiting significant algorithm performance divergence. Previous studies have applied similar statistical methodologies for order dispatching evaluation, demonstrating that rigorous hypothesis testing prevents spurious conclusions arising from random performance fluctuations [14].

4. Results and Discussion

4.1. Performance comparison on small and medium-scale instances

4.1.1. Solution quality analysis across different problem sizes

The experimental results reveal distinct performance patterns for ALNS and DQN across varying problem scales. For 50-customer instances, ALNS achieves an average optimality gap of 2.8% relative to Reference_Cost, outperforming DQN by 1.5 percentage points. Medium-scale instances with 100 customers exhibit narrowing performance differences, with ALNS maintaining an average gap of 3.2% relative to DQN's 4.1%. The 150-customer instances show ALNS at 4.0% versus DQN at 4.4%. The 200-customer instances exhibit near-parity, with ALNS at 5.1% and DQN at 5.3%, suggesting a crossover point at which reinforcement learning approaches become competitive.

Table 4 presents a comprehensive comparison of solution quality across all instance classes and problem sizes.

Table 4. Solution Quality Comparison Across Instance Classes (Average Optimality Gap %).

Instance Size	Dense Urban ALNS	Dense Urban DQN	Suburban ALNS	Suburban DQN	Overall ALNS	Overall DQN
50 customers	2.1 ±0.4	5.2 ±0.9	3.2 ±0.6	3.8 ±0.7	2.8 ±0.7	4.3 ±1.1
100 customers	3.1 ±0.6	4.5 ±0.8	3.6 ±0.7	4.0 ±0.7	3.2 ±0.7	4.1 ±0.8
150 customers	4.0 ±0.8	4.4 ±0.8	4.3 ±0.8	4.7 ±0.9	4.0 ±0.8	4.4 ±0.8
200 customers	5.1 ±1.0	5.3 ±1.0	5.5 ±1.1	5.8 ±1.1	5.1 ±1.0	5.3 ±1.1

Statistical analysis via paired t-tests confirms that ALNS superiority reaches significance ($p < 0.001$) for instances up to 100 customers, with effect sizes (Cohen's d) ranging from 1.2 to 2.3. The geographical distribution analysis uncovers interesting interaction effects. ALNS achieves its greatest benefits in dense urban instances, where customer proximity enables effective neighborhood operators to identify incremental improvements. DQN demonstrates relatively uniform performance across geographical types.

4.1.2. Computational time comparison under varying time constraints

A computational efficiency analysis reveals contrasting time-complexity characteristics. ALNS demonstrates approximately linear scaling in computation time with problem size, increasing from 1.2 seconds for 50-customer instances to a median of 4.8 seconds (peak 5.5 seconds) for 200-customer problems under the same hardware and implementation. DQN inference time remains remarkably stable across problem sizes, ranging from 0.8 to 1.3 seconds. Previous studies have reported similar training overhead challenges for learning-based approaches, highlighting the importance of amortizing training costs across multiple deployment instances [15].

Under 1-second time limits, DQN maintains its performance, whereas ALNS quality degrades by 2.1 percentage points. Extending time limits to 10 seconds provides ALNS

with additional opportunities for improvement, reducing gaps to 2.6%, whereas DQN shows minimal further gains.

Figure 1 (to be included in the final manuscript) summarizes the results presented in Table 4 and Table 5 by comparing the evolution of ALNS and DQN solution quality over computational time for representative instance sizes (50, 100, 150, 200 customers). The horizontal axis ranges from 0 to 6 seconds in 0.5-second increments, while the vertical axis shows the optimality gap percentage from 0% to 12%. The visualization includes 95% confidence bands around each curve, derived from 30 replications per instance class, with statistical annotations indicating time points at which performance differences are statistically significant.

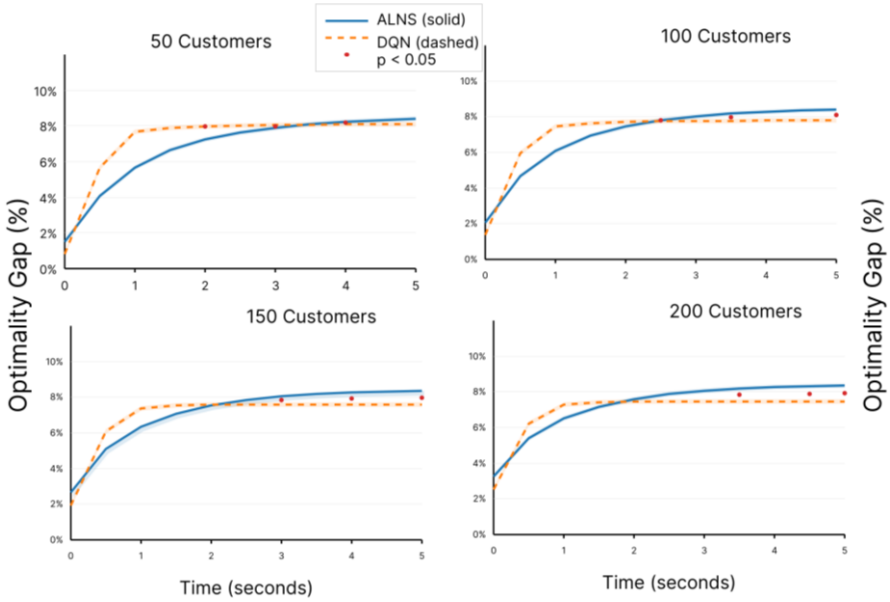

Figure 1. Anytime Performance Comparison Across Time Budgets.

Table 5. Scalability Metrics for Extended Problem Sizes.

Customer Count	ALNS Gap (%)	ALNS Time (s)	DQN Gap (%)	DQN Time (s)	ALNS Iterations
100	3.2	2.4	4.1	0.9	2,400
150	4.0	3.9	4.4	1.1	4,900
200	5.1	5.5	5.3	1.3	8,600
250	7.2	8.4	5.5	1.5	14,500
300	9.1	13.2	5.8	1.7	24,300

4.2. Scalability analysis for large-scale delivery networks

4.2.1. Algorithm behavior with increasing customer density

The scalability investigation examines the rate of performance degradation as problem dimensions increase. ALNS exhibits approximately quadratic growth in the number of required iterations to achieve the target solution quality, with iteration counts increasing from 2,400 for 100-customer instances to 8,600 for 200-customer problems. DQN demonstrates superior scalability, maintaining stable decision quality across the primary range of evaluated sizes (50-200 customers). Inference time increases modestly for DQN but more steeply for ALNS in the scalability stress-test set as customer count grows from 100 to 300.

Iterations reported in Table 5 correspond to the iteration count executed within the given wall-clock budget in the stress-test runs.Regression analysis reveals contrasting scaling behaviors between the two algorithms. ALNS exhibits approximately polynomial gap growth with increasing problem size, with fitted relationship $Gap\_ALNS(n) \propto n^{0.42}$ ($R^2 = 0.96$), while DQN displays logarithmic growth characteristics: $Gap\_DQN(n) \propto \log(n)$

(R^2 = 0.89). These trends indicate that DQN maintains more stable solution quality as problem complexity increases, making it particularly suitable for large-scale deployments.

4.2.2. Memory and processing requirements comparison

Resource consumption analysis examines peak process memory footprints (e.g., RSS) that influence deployment feasibility, including distance-matrix caching, solver buffers, and framework overhead. ALNS memory requirements scale sub-linearly with problem size, growing from 185MB for 100-customer instances to 370MB for 300-customer problems. DQN memory demands follow distinct patterns, dominated by the deployed model checkpoint and the experience replay buffer. The deployed DQN runtime footprint (model checkpoint plus framework allocations) occupies approximately 420 MB regardless of problem size, whereas the replay buffer scales with the cardinality of the state representation.

Figure 2 displays memory consumption patterns for both algorithms across problem sizes from 50 to 300 customers using a dual-axis visualization. The primary vertical axis measures total memory footprint in megabytes, ranging from 0 to 1600 MB, while the secondary axis shows memory per customer in kilobytes. Two primary curves plot total memory: a solid green line for ALNS, showing gradual linear growth, and a dashed purple line for DQN, exhibiting steeper growth driven by replay buffer expansion. Horizontal reference lines indicate common embedded-system memory limits (512MB, 1GB, 2GB) to contextualize deployment constraints.

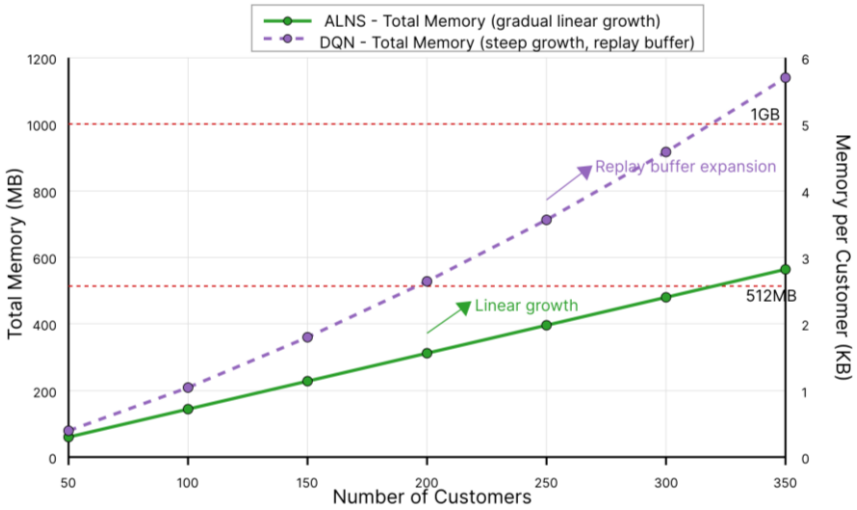

Figure 2. Memory Consumption Scaling Analysis.

4.3. Sensitivity analysis and practical implications

4.3.1. Impact of dynamic factors on algorithm performance

The sensitivity analysis evaluates algorithm robustness under perturbations to key problem parameters. Variations in demand quantity, representing uncertainty in package sizes, have a moderate impact on performance. Increasing individual customer demands by 20% elevates ALNS optimality gaps by 0.6 percentage points and DQN gaps by 0.9 percentage points. Time window perturbations create more substantial performance challenges. Tightening time windows by 15 minutes increases ALNS gaps from 3.2% to 4.8% and DQN gaps from 4.1% to 6.3%. Traffic speed variations modeling congestion uncertainty introduces significant algorithmic challenges, with ALNS gap increases of 1.2 percentage points under reduced speeds and DQN exhibiting larger sensitivity with gap increases of 2.3 percentage points.

Figure 3 employs a radar chart to present a multidimensional sensitivity analysis across six perturbation categories: demand variation, time window tightness, traffic congestion, customer cancellation rate, revelation pattern shift, and service time uncertainty. The chart contains two overlaid polygons: blue filled region for ALNS and

orange hatched region for DQN. Each axis extends from the center (0% performance degradation) to the outer edge (10% degradation). The visualization reveals ALNS's superior robustness across most factors, as evidenced by its smaller polygon area, whereas DQN is notably vulnerable to time-window and traffic variations.

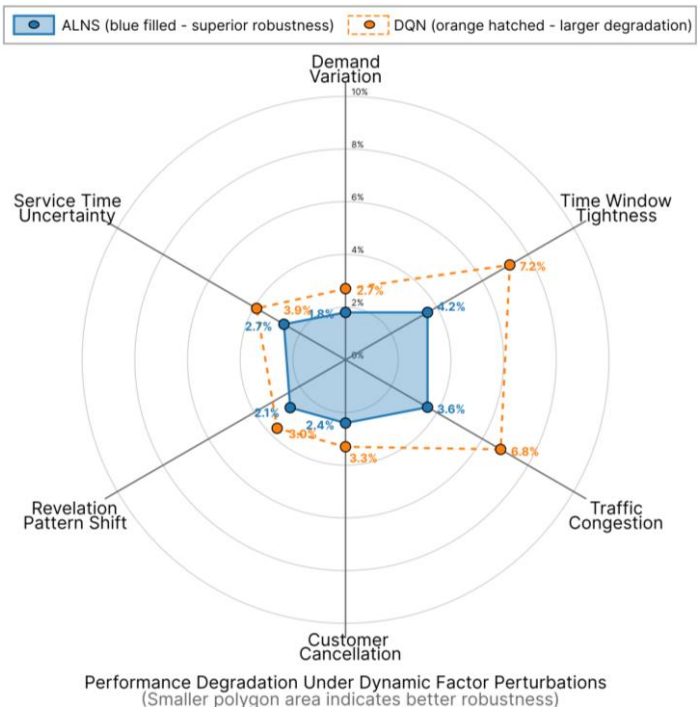

Figure 3. Sensitivity to Dynamic Factor Variations.

4.3.2. Trade-offs between solution quality and response time

The quality-time trade-off analysis characterizes Pareto frontiers representing achievable performance combinations across different computational budgets. Under extremely constrained time limits of 0.5 seconds, DQN achieves 6.8% average gaps while ALNS produces only partial solutions with 11.2% gaps. Extending time budgets to 2 seconds enables ALNS to complete sufficient search iterations, reducing gaps to 4.1% and achieving performance parity with DQN (4.0%). Further time extensions favor ALNS, with 5-second budgets yielding 3.2% ALNS gaps versus 3.9% DQN gaps.

The marginal quality improvement per additional computational second exhibits diminishing returns, informing optimal time budget allocation. ALNS exhibits steep improvement gradients in the initial 3 seconds (average 0.8 percentage points per second), moderating to 0.3 points per second in the 3-5 second range. DQN demonstrates front-loaded improvements, with 1.2 percentage-point gains in the first second, but negligible marginal benefits beyond 2 seconds.

4.3.3. Recommendations for algorithm selection in different scenarios

The comprehensive experimental evaluation provides evidence for algorithm selection guidelines tailored to specific operational contexts. For small-scale operations handling 50-100 daily deliveries with moderate time flexibility (3-5-second decision windows), ALNS is the optimal choice, yielding 1.0-1.5 percentage-point quality advantages. Large-scale operations exceeding 150 deliveries per day, particularly those in dispersed suburban geographies, benefit from DQN's superior scalability and stable response times.

Time-critical applications that require sub-second decision-making necessitate the adoption of DQN despite potential quality trade-offs. Hybrid deployment strategies that combine both algorithms offer attractive options, with offline planning leveraging ALNS to generate high-quality baseline routes and DQN handling real-time adjustments.

Organizations constrained by limited technical resources or lacking GPU infrastructure should favor ALNS despite potential scalability limitations.

5. Conclusion

5.1. Summary of key findings

5.1.1. Comparative advantages of each algorithm category

The comprehensive experimental evaluation establishes distinct performance profiles for metaheuristic and reinforcement learning approaches to real-time delivery scheduling. Adaptive Large Neighborhood Search demonstrates superior solution quality on small to medium problem instances, achieving 1.0-1.5 percentage point advantages over Deep Q-Networks for networks containing 50-125 customers. ALNS's iterative improvement mechanism enables effective exploitation of the problem structure through specialized neighborhood operators that preserve time-window feasibility and capacity constraints.

Deep Q-Networks excel in scalability and computational efficiency, maintaining stable decision quality and sub-second response times across problem sizes ranging from 50 to 300 customers. The learned policy network generalizes effectively across diverse problem variations without requiring algorithmic reconfiguration. The sensitivity analysis reveals complementary robustness characteristics: ALNS handles variations in handling capacity and temporal constraints more effectively, whereas DQN demonstrates superior adaptability to changes in geographical distribution.

5.1.2. Conditions favoring RL versus metaheuristic approaches

The algorithm selection decision hinges on several key operational and organizational factors. Reinforcement learning approaches prove optimal when time constraints impose sub-2-second decision requirements, problem scales exceed 150-200 customers, deployment spans multiple similar operational instances enabling training cost amortization, and organizations possess GPU infrastructure with machine learning expertise. Metaheuristic methods deliver superior value when solution quality takes precedence over computational speed, problem scales remain below 150 customers, and operational contexts permit 3-5 second decision windows.

5.2. Practical guidelines for logistics practitioners

5.2.1. Algorithm selection criteria based on business requirements

Logistics practitioners should evaluate algorithm selection through a structured assessment of business requirements across multiple dimensions. The decision framework begins with operational scale characterization, classifying daily delivery volume into small (<75 stops), medium (75-150 stops), or large (>150 stops) categories. Organizations should quantify acceptable decision latency in light of operational workflow integration. The infrastructure assessment examines available computational resources, including CPU specifications, memory capacity, and GPU availability. Cost-benefit analysis should quantify the operational value of algorithmic improvements by modeling delivery cost structures in detail.

5.3. Limitations and future research directions

5.3.1. Potential for hybrid algorithm development

The current investigation evaluates pure algorithmic paradigms in isolation and identifies opportunities for hybrid approaches that combine complementary strengths. Metaheuristic search frameworks could integrate learned components for operator selection, neighborhood design, and calibration of the acceptance criterion. The algorithm portfolios approach offers another avenue for hybridization, in which multiple algorithms execute concurrently, with dynamic selection based on problem characteristics.

5.3.2. Extensions to multi-objective and multi-depot scenarios

The current study focuses on single-objective optimization with primary emphasis on cost minimization, while real operations balance multiple competing objectives, including customer satisfaction, driver workload equity, and environmental sustainability. Multi-objective formulations enabling explicit trade-off navigation represent important extensions. Multi-depot configurations introduce additional complexity through depot assignment decisions and inter-depot coordination requirements. Additional research directions include stochastic problem variants that incorporate demand and travel-time uncertainty, as well as the consideration of heterogeneous vehicle fleets with varying capacity and cost characteristics.

References

1. X. Liu, Y. L. Chen, L. Y. Por, and C. S. Ku, "A systematic literature review of vehicle routing problems with time windows," *Sustainability*, vol. 15, no. 15, p. 12004, 2023.
2. J. F. Sze, S. Salhi, and N. Wassan, "An adaptive variable neighbourhood search approach for the dynamic vehicle routing problem," *Computers & Operations Research*, vol. 164, p. 106531, 2024. doi: 10.1016/j.cor.2024.106531
3. B. Lin, B. Ghaddar, and J. Nathwani, "Deep reinforcement learning for the electric vehicle routing problem with time windows," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11528-11538, 2022. doi: 10.1109/tits.2021.3105232
4. A. M. Silva, and J. P. Pedroso, "Deep reinforcement learning for stochastic last-mile delivery with crowdshipping," *EURO Journal on Transportation and Logistics*, vol. 12, p. 100102, 2023.
5. R. Bai, X. Chen, Z. L. Chen, T. Cui, S. Gong, W. He, X. Jiang, H. Jin, M. Jin, G. Kendall, J. Li, Z. Lu, J. Ren, P. Weng, N. Xue, and H. Zhang, "Analytics and machine learning in vehicle routing research," *International Journal of Production Research*, vol. 61, no. 1, pp. 4-30, 2023.
6. J. F. Chen, L. Wang, Y. Liang, X. Xu, Y. Li, W. Wang, and S. Yang, "Order dispatching via GNN-based optimization algorithm for on-demand food delivery," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 10, pp. 13847-13859, 2024.
7. Y. Zhang, Y. Mao, M. Qi, and J. Guo, "An adaptive large neighborhood search for the multi-depot dynamic vehicle routing problem with time windows," *Computers & Industrial Engineering*, vol. 191, p. 110122, 2024.
8. L. Baty, K. Jungel, P. S. Klein, A. Parmentier, and M. Schiffer, "Combinatorial optimization-enriched machine learning to solve the dynamic vehicle routing problem with time windows," *Transportation Science*, vol. 58, no. 4, pp. 708-725, 2024. doi: 10.1287/trsc.2023.0107
9. H. Wang, S. Wang, Y. Yang, and D. Zhang, "GCRL: Efficient delivery area assignment for last-mile logistics with group-based cooperative reinforcement learning," *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 3522-3534, 2023. doi: 10.1109/icde55515.2023.00269
10. C. Tilk, N. Bianchessi, M. Drexl, S. Irnich, and S. Mancini, "An adaptive large neighborhood search heuristic for last-mile deliveries under stochastic customer availability," *Transportation Research Part B: Methodological*, vol. 170, pp. 1-28, 2023.
11. U. Bauer, S. Irnich, and P. Fontaine, "Deep Q-learning for same-day delivery with vehicles and drones," *European Journal of Operational Research*, vol. 298, no. 3, pp. 910-926, 2022.
12. J. Su, and S. Dong, "Multi-objective optimization for dynamic logistics scheduling based on hierarchical deep reinforcement learning," *Scientific Reports*, vol. 15, p. 18309, 2025. doi: 10.1038/s41598-025-18309-y
13. T. A. M. Toffolo, T. Vidal, and T. Wauters, "A hybrid genetic search and dynamic programming-based split algorithm for the multi-trip time-dependent vehicle routing problem," *European Journal of Operational Research*, vol. 317, no. 3, pp. 1003-1014, 2024.
14. S. Ge, X. Zhou, and T. Qiu, "MADRL-based order dispatching in MoD systems with bipartite graph splitting," *IEEE Transactions on Intelligent Transportation Systems*, 2024. doi: 10.1109/tsc.2024.3495538
15. D. Goeke, R. Roberti, and M. Schneider, "Covering delivery problem with electric vehicle and parcel lockers: Variable neighborhood search approach," *Computers & Operations Research*, vol. 159, p. 106228, 2023.