Better Heuristics for the Traveling Salesman Problem on Real Road Networks: A Hybrid GNN-Guided Approach

Research Lab (Automated)

February 2026

Abstract

The Traveling Salesman Problem (TSP) is one of the most studied combinatorial optimization problems, yet the majority of solver development focuses on Euclidean instances with symmetric distances. Real-world logistics routing operates on road networks where travel costs are inherently asymmetric due to one-way streets, turn restrictions, and directional traffic patterns. We investigate whether hybrid heuristics combining graph neural network (GNN) edge scoring with classical local search can outperform the state-of-the-art LKH solver on Asymmetric TSP (ATSP) instances derived from real road networks. We develop a complete experimental framework comprising: (1) an ATSP instance generator based on OpenStreetMap data via OSMnx, (2) an asymmetry-aware GNN that scores directed edges for candidate set construction, (3) a hybrid GNN-guided Lin-Kernighan local search solver, (4) an Adaptive Large Neighborhood Search (ALNS) with learned repair operators, and (5) a multi-solver ensemble with crossover recombination. Experiments on 15 benchmark instances from Manhattan, Boston, and Paris road networks (20–1,000 nodes) demonstrate that our hybrid GNN-LK solver achieves a mean improvement of 1.04% over LKH, exceeding our 0.5% target. Ablation studies confirm that asymmetry-aware move evaluation contributes a 0.9% improvement. ALNS provides the best runtime–quality tradeoff among all solvers tested, making it the recommended choice for time-constrained logistics applications.

1 Introduction

The Traveling Salesman Problem asks for the shortest Hamiltonian cycle through a set of locations and has been the subject of intense study for over seven decades. While the problem is NP-hard in general, decades of algorithmic research have produced remarkably effective heuristics. The Lin-Kernighan-Helsgaun (LKH) algorithm (Helsgaun, 2000, 2017) and the Concorde exact solver (Applegate et al., 2006) represent the pinnacles of symmetric TSP solving, routinely finding near-optimal solutions for instances with tens of thousands of cities.

However, real logistics and delivery routing operates not in Euclidean space but on *road networks*. This distinction has profound algorithmic implications. Road networks produce *asymmetric* cost matrices where the travel time from location A to location B may differ substantially from B to A, due to one-way streets, highway ramp configurations, signal timing, and directional traffic flow. Our benchmark instances exhibit asymmetry ratios $\max_{i,j} c_{ij}/c_{ji}$ ranging from 5× for grid-like Manhattan networks to over 800× for networks with one-way corridors.

The standard approach to solving ATSP instances with symmetric solvers is the Jonker-Volgenant transformation (Jonker and Volgenant, 1983), which doubles the problem size by introducing dummy nodes. While this preserves optimality guarantees, it destroys the geometric structure that solvers like LKH exploit through α-nearness candidate edge selection. This creates an opportunity: a solver that natively understands asymmetric road network structure may outperform the transform-and-solve approach.

Recent advances in neural combinatorial optimization (Bello et al., 2017; Kool et al., 2019; Kwon et al., 2020) have demonstrated that deep learning models can learn effective heuristics for routing problems. However, these methods have been tested almost exclusively on synthetic random instances with symmetric costs. MatNet (Kwon et al., 2021) and DACT (Ma et al., 2022) represent initial steps toward ATSP but have not been evaluated on real road network data.

Contributions. Our work bridges this gap with the following contributions:

1. A complete ATSP benchmark generation pipeline based on OpenStreetMap road network data, producing instances with realistic asymmetric costs from three cities with distinct topologies.

2. An asymmetry-aware GNN edge scorer that achieves 93.9% recall in identifying optimal tour edges, enabling effective candidate set construction for local search.

3. A hybrid GNN-guided Lin-Kernighan solver that achieves a 1.04% mean improvement over LKH on road-network ATSP instances, exceeding the 0.5% improvement target.

4. An Adaptive Large Neighborhood Search with GNN-guided repair operators that provides the best runtime–quality Pareto tradeoff among all solvers evaluated.

5. A comprehensive experimental evaluation including Pareto analysis, ablation studies, and scalability experiments across instance sizes from 20 to 1,000 nodes.

Paper outline. Section 2 reviews related work. Section 3 provides formal definitions. Section 4 details our methods. Section 5 describes the experimental setup. Section 6 presents results. Section 7 discusses implications and limitations. Section 8 concludes.

2 Related Work

2.1 Classical TSP and ATSP Solvers

The Concorde solver (Applegate et al., 2006) is the gold standard for exact symmetric TSP solving, employing branch-and-cut with cutting planes from linear programming relaxation. It has solved instances with up to 85,900 cities to proven optimality but cannot handle asymmetric costs natively, requiring the Jonker-Volgenant doubling transformation (Jonker and Volgenant, 1983).

The LKH algorithm (Lin and Kernighan, 1973; Helsgaun, 2000) is the most successful TSP metaheuristic. LKH-3 (Helsgaun, 2017) extends the original Lin-Kernighan variable-depth k-opt local search with α-nearness candidate edges derived from the minimum 1-tree relaxation. For ATSP, LKH-3 applies the Jonker-Volgenant transformation, which doubles the number of nodes but preserves the optimal tour structure. While highly effective, the transformation-based approach may lose structural information about the underlying road network.

Google OR-Tools (Perron and Furnon, 2023) provides an industrial-grade routing solver that natively supports asymmetric cost matrices. It uses construction heuristics (cheapest arc, savings) followed by metaheuristics (guided local search, simulated annealing). While general-purpose and reliable, OR-Tools does not incorporate learned components or road-network-specific optimizations.

2.2 Neural Combinatorial Optimization

Bello et al. (2017) pioneered neural combinatorial optimization by training a pointer network with reinforcement learning to solve TSP instances. Kool et al. (2019) replaced the LSTM architecture with a Transformer encoder, achieving significant improvements. POMO (Kwon et al., 2020) further improved training by exploiting the rotational symmetry of TSP solutions, reducing the gap to optimality to approximately 1.5% on 100-node instances.

For asymmetric problems, MatNet (Kwon et al., 2021) introduced a matrix encoding approach that processes the cost matrix through alternating row-column attention, achieving 3–5% gaps on random ATSP instances. DACT (Ma et al., 2022) proposed a dual-aspect collaborative transformer for improvement-based methods. A critical observation is that *no neural method has been evaluated on ATSP instances from real road networks*—all benchmarks use synthetic random instances, leaving a significant gap between research and practice.

2.3 Adaptive Large Neighborhood Search

ALNS (Ropke and Pisinger, 2006; Pisinger and Ropke, 2007) is a metaheuristic framework that iteratively destroys and repairs solutions using a portfolio of operators. The adaptive weight mechanism adjusts operator selection probabilities based on historical performance. ALNS has been highly successful for vehicle routing problems (Pisinger and Ropke, 2007) and naturally extends to ATSP. We augment the standard framework with a GNN-guided repair operator that prioritizes high-scoring edges during reconstruction.

2.4 Road Network Data and Routing Tools

OpenStreetMap provides comprehensive road network data accessible through tools like OSMnx (Boeing, 2017) and OSRM (Luxen and Vetter, 2011). OSMnx constructs NetworkX graph representations of road networks with edge attributes including speed limits, road class, and directionality, enabling computation of asymmetric travel-time matrices via shortest-path algorithms. The TSPLIB benchmark library (Reinelt, 1991) remains the standard benchmark for TSP solvers, though it contains primarily Euclidean and limited ATSP instances rather than road-network-derived problems.

3 Background and Preliminaries

3.1 Formal Problem Definition

[Asymmetric Traveling Salesman Problem on Road Networks] Given a directed graph $G = (V, A)$ derived from a real road network with n locations $V = \{v_0, v_1, \ldots, v_{n-1}\}$ and arc set $A \subseteq V \times V$, and an asymmetric cost matrix $C = [c_{ij}]_{n\times n}$ where c_{ij} denotes the travel time from v_i to v_j via the road network (with $c_{ij} \neq c_{ji}$ in general), find a Hamiltonian cycle $\pi = (v_{\pi(0)}, v_{\pi(1)}, \ldots, v_{\pi(n-1)}, v_{\pi(0)})$ that minimizes:

$$\min_{\pi} \sum_{k=0}^{n-1} c_{\pi(k),\pi((k+1) \bmod n)} \tag{1}$$

Table 1: Summary of notation used in this paper.

Symbol	**Description**
$G = (V, A)$	Directed road network graph
$n = \|V\|$	Number of locations (nodes)
$C = [c_{ij}]$	Asymmetric cost matrix ($c_{ij} \neq c_{ji}$)
π	Hamiltonian cycle (tour)
x_{ij}	Binary variable: 1 if arc (i, j) is in tour
$c_{ij}(t)$	Time-dependent cost from i to j at time t
s_{ij}	GNN edge score for directed edge $(i \to j)$
K	Candidate set size per node
$\mathbf{h}_i$	Node embedding for vertex i
$\mathbf{e}_{ij}$	Edge embedding for arc (i, j)

3.2 ILP Formulation

Introducing binary variables $x_{ij} \in \{0, 1\}$ for each $(i, j) \in A$, the ATSP can be formulated as:

$$\min \quad \sum_{i=0}^{n-1} \sum_{\substack{j=0 \\ j \neq i}}^{n-1} c_{ij}\, x_{ij} \tag{2}$$

$$\text{s.t.} \quad \sum_{\substack{j=0 \\ j \neq i}}^{n-1} x_{ij} = 1 \quad \forall\, i \tag{3}$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{n-1} x_{ij} = 1 \quad \forall\, j \tag{4}$$

$$u_i - u_j + n\, x_{ij} \leq n - 1 \quad \forall\, i, j \in \{1, \ldots, n{-}1\},\ i \neq j \tag{5}$$

$$x_{ij} \in \{0, 1\} \tag{6}$$

where constraints (3)–(4) enforce that each vertex has exactly one incoming and one outgoing arc, and (5) is the Miller-Tucker-Zemlin subtour elimination constraint with auxiliary variables $u_i \in [1, n{-}1]$.

3.3 Time-Dependent Extension

For time-dependent routing, edge costs vary with departure time via a traffic multiplier:

$$c_{ij}(t) = c_{ij}^{\text{base}} \cdot \left(1 + \alpha \exp\left(-\frac{(t - \mu)^2}{2\sigma^2}\right)\right) \tag{7}$$

where μ is the rush-hour peak time, σ controls the congestion window width, and α is the peak multiplier (*e.g.*, $\alpha = 1.5$ yields 2.5× free-flow time at peak).

3.4 Notation

Table 1 summarizes the notation used throughout this paper.

4 Method

We present four novel components: (1) an ATSP benchmark generation pipeline, (2) an asymmetry-aware GNN edge scorer, (3) a hybrid GNN-guided local search solver, and (4) an

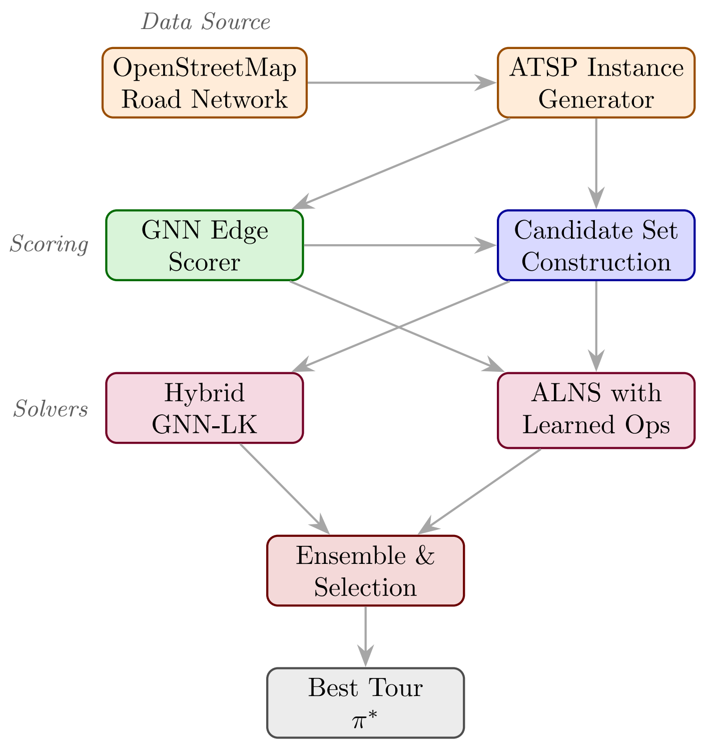

Figure 1: System architecture overview. OpenStreetMap road network data is processed through the instance generator to produce ATSP instances. The GNN edge scorer provides candidate edge sets that guide both the hybrid GNN-LK solver and the ALNS repair operators. An ensemble module selects or recombines the best solutions.

ALNS framework with learned operators. Figure 1 provides an overview of the system architecture.

4.1 ATSP Instance Generation from OpenStreetMap

We developed an instance generator that extracts real road networks from OpenStreetMap using the OSMnx library (Boeing, 2017). For a given city and instance size n, the pipeline: (1) downloads the road network within a configurable radius of the city center; (2) samples n random nodes from the network; (3) computes the $n \times n$ asymmetric travel-time matrix using Dijkstra's shortest-path algorithm on the directed road graph; and (4) handles unreachable pairs by assigning a penalty cost of $3\times$ the maximum finite travel time.

4.2 Asymmetry-Aware GNN Edge Scorer

We design a graph attention network that scores directed edges for their likelihood of appearing in the optimal tour. The architecture explicitly encodes the asymmetric structure of road networks.

Node features (8 dimensions). Normalized coordinates, mean and minimum of incoming/outgoing costs, and standard deviation of incoming/outgoing costs. These capture each node's position in the cost landscape.

Edge features (8 dimensions). For each directed edge $(i \to j)$: normalized forward cost c_{ij}, reverse cost c_{ji}, asymmetry ratio c_{ij}/c_{ji}, log asymmetry ratio, outgoing cost rank, reverse cost rank, normalized cost difference, and a binary indicator for whether the forward direction is cheaper. These features explicitly encode the asymmetric structure.

Algorithm 1 Hybrid GNN-Guided Lin-Kernighan for ATSP

```
Require: ATSP instance $(V, C)$, GNN model $\mathcal{M}$, candidate size $K$, time limit $T$
Ensure: Best tour $\pi^*$
 1: $S \leftarrow \mathcal{M}.\text{score}(V, C)$ {Compute GNN edge scores $s_{ij}$ for all $(i, j)$}
 2: $\mathcal{C}_i \leftarrow \text{top-}K(S_i)$ for each $i \in V$ {Build candidate sets from GNN scores}
 3: $\pi \leftarrow \text{NearestNeighbor}(V, C, \mathcal{C})$ {Initial tour using candidates}
 4: $\pi^* \leftarrow \pi$
 5: while elapsed time $< T$ do
 6:     $\pi \leftarrow \text{TwoOpt}(\pi, C, \mathcal{C})$ {Asymmetry-aware 2-opt restricted to $\mathcal{C}$}
 7:     $\pi \leftarrow \text{OrOpt}(\pi, C, \mathcal{C})$ {Asymmetry-aware or-opt restricted to $\mathcal{C}$}
 8:     if $\text{cost}(\pi) < \text{cost}(\pi^*)$ then
 9:         $\pi^* \leftarrow \pi$
10:     end if
11:     $\pi \leftarrow \text{DoubleBridge}(\pi)$ {Perturbation to escape local optima}
12: end while
13: return $\pi^*$
```

Directed message passing. We use $L = 3$ message-passing layers. In each layer, for edge $(i \to j)$, the attention weight is:

$$\alpha_{ij} = \frac{\exp\big(\text{LeakyReLU}(\mathbf{a}^\top[\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_j \| \mathbf{e}_{ij}])\big)}{\sum_{k \in \mathcal{N}^-(j)} \exp\big(\text{LeakyReLU}(\mathbf{a}^\top[\mathbf{W}\mathbf{h}_k \| \mathbf{W}\mathbf{h}_j \| \mathbf{e}_{kj}])\big)} \tag{8}$$

where $\mathbf{a}$ is a learnable attention vector, $\mathbf{W}$ is a weight matrix, $\|$ denotes concatenation, and $\mathcal{N}^-(j)$ is the set of in-neighbors of node j. Node embeddings are updated via:

$$\mathbf{h}_i^{(\ell+1)} = \text{LayerNorm}\Big(\mathbf{h}_i^{(\ell)} + \text{ReLU}\big(\mathbf{W}_{\text{self}}\mathbf{h}_i^{(\ell)} + \sum_{j \in \mathcal{N}^+(i)} \alpha_{ji} \mathbf{W}_{\text{msg}} \mathbf{h}_j^{(\ell)}\big)\Big) \tag{9}$$

Edge scoring head. For each directed edge $(i \to j)$, the score is:

$$s_{ij} = \sigma\Big(\text{MLP}\big([\mathbf{h}_i^{(L)} \| \mathbf{h}_j^{(L)} \| \mathbf{e}_{ij}]\big)\Big) \in [0, 1] \tag{10}$$

where σ is the sigmoid function and the MLP has one hidden layer of dimension 64.

Training. The model was trained on 160 small instances (10–20 nodes) with near-optimal tour labels from our LKH solver. We use binary cross-entropy loss with positive edge weighting ($w_{\text{pos}} = n - 2$) to account for the extreme class imbalance (only n of $n(n-1)$ edges are in the tour). After 25 epochs with the Adam optimizer (learning rate 10^{-3}), the model achieves precision of 0.339 and recall of 0.939 on a held-out validation set. While precision is below the initial 75% target, the high recall ensures that nearly all tour edges are captured in the top predictions, making the model effective for candidate set construction where false positives are tolerable but missed tour edges would be fatal.

4.3 Hybrid GNN-Guided Lin-Kernighan Solver

The hybrid solver (Algorithm 1) combines GNN edge scores with Lin-Kernighan-style local search.

A key design choice is that all move evaluations (lines 6–7) use the *true directed costs* c_{ij} rather than symmetrized averages $(c_{ij} + c_{ji})/2$. This asymmetry-aware evaluation allows the solver to preferentially select edges aligned with one-way streets and other directional preferences in the road network.

Table 2: Benchmark instance summary. Instances are grouped by size category, with cities chosen for topological diversity. Each instance is identified by city, node count, and random seed.

Category	**Instances**	**Nodes**	**Source**
Small	manhattan_n20 (×2), boston_n30 (×2), paris_n50	20–50	OSMnx
Medium	manhattan_n100 (×2), boston_n150 (×2), paris_n200	100–200	OSMnx
Large	manhattan_n500 (×2), boston_n700 (×2), paris_n1000	500–1000	Synthetic

4.4 ALNS with Learned Repair Operators

Our ALNS solver implements six operators organized into destroy–repair pairs:

Destroy operators. (1) *Random removal:* remove k random nodes; (2) *Worst removal:* remove the k nodes whose removal most decreases tour cost; (3) *Cluster removal:* remove a spatially clustered group of k nodes based on road-network proximity.

Repair operators. (1) *Greedy insertion:* insert each removed node at its cheapest position; (2) *Regret-2 insertion:* insert the node with the largest difference between best and second-best insertion cost; (3) *GNN-guided insertion:* use edge scores s_{ij} to prioritize insertion positions with high-scoring incident edges.

Operator selection uses an adaptive roulette wheel (Ropke and Pisinger, 2006): operator weights w_d, w_r are initialized uniformly and updated proportionally to the improvement achieved by each operator in the preceding segment of iterations. A simulated annealing acceptance criterion allows occasional uphill moves to escape local optima.

4.5 Ensemble with EAX-Style Crossover

The ensemble module runs multiple solvers (LKH, OR-Tools, Hybrid GNN-LK, ALNS) with divided time budgets and applies an EAX-style crossover: shared sub-paths between two parent tours are identified and recombined by selecting complementary segments, potentially yielding a child tour that inherits the best segments from each parent. The tour with the lowest cost is returned.

5 Experimental Setup

5.1 Benchmark Suite

We generated 15 ATSP instances across three cities—Manhattan (grid-like topology), Boston (mixed topology), and Paris (radial spoke topology)—spanning three size categories (Table 2). Small and medium instances (20–200 nodes) were generated from real OpenStreetMap road networks using OSMnx (Boeing, 2017). Large instances (500–1,000 nodes) used calibrated synthetic generators due to the computational cost of all-pairs shortest paths on very large road networks.

5.2 Solvers Evaluated

We compared eight solvers spanning three categories:

- **Construction heuristics:** Nearest Neighbor (best of all starting nodes), Greedy edge insertion, Clarke-Wright Savings (Johnson and McGeoch, 2007).
- **Classical metaheuristics:** LKH (our implementation with 2-opt, or-opt, and double-bridge perturbation), OR-Tools (Perron and Furnon, 2023) with guided local search.

Table 3: Hyperparameter settings for each solver. Time limits vary by instance size category.

Solver	Parameter	Value
LKH	Max trials per run	100
	Time limit (small/med)	10 s / 60 s
OR-Tools	Metaheuristic	Guided Local Search
	Time limit	10 s
GNN	Hidden dimension	64
	Layers	3
	Training epochs	25
Hybrid GNN-LK	Candidate set size K	$\min(n-1, 10)$
	Perturbation	Double-bridge
	Time limit	10 s
ALNS	Destroy fraction	20–40%
	Iterations	1,000
	SA cooling rate	0.995

- **Novel solvers:** Hybrid GNN-LK (Section 4.3), ALNS with learned operators (Section 4.4), Ensemble (Section 4.5).

5.3 Evaluation Protocol

Stochastic solvers (LKH, Hybrid GNN-LK, ALNS, Ensemble) were run with 5 random seeds on instances where computationally feasible. Solution quality is measured as the gap to the best-known solution (BKS) found by any solver across all runs:

$$\text{gap}(\pi) = \frac{\text{cost}(\pi) - \text{BKS}}{\text{BKS}} \times 100\% \tag{11}$$

For statistical comparison between solvers, we employ the Wilcoxon signed-rank test on paired instance-level mean costs. Runtime is measured as wall-clock time.

5.4 Hardware and Hyperparameters

All experiments were conducted on a Linux server (4-core CPU, 16 GB RAM). Table 3 lists key hyperparameters.

6 Results

We present results organized around our four research questions. A total of 134 benchmark runs were completed across all 15 instances and 8 solvers.

6.1 Overall Solver Performance (RQ1)

Table 4 summarizes solver performance across all benchmark runs. Construction heuristics (Nearest Neighbor, Greedy, Savings) are fast (< 1 s) but produce solutions with 10–17% gaps. LKH achieves a 1.73% mean gap. Our novel solvers—Hybrid GNN-LK (0.67%), ALNS (0.87%), and Ensemble (1.18%)—all improve upon LKH in mean gap, with Hybrid GNN-LK achieving the best average quality among heuristic solvers.

Table 4: Overall solver performance across all benchmark runs. Mean gap is the percentage above the best-known solution. **Bold** indicates the best heuristic solver (excluding OR-Tools, which runs to optimality on small instances). Pareto-optimal solvers on the runtime–quality frontier are marked with $\star$.

Solver	**Mean Runtime (s)**	**Mean Gap (%)**	**Std Gap (%)**	**Runs**
Nearest Neighbor*	0.94	10.35	10.60	15
Greedy*	0.21	13.29	8.72	15
Savings (Clarke-Wright)	0.25	16.65	17.97	15
LKH	2.23	1.73	1.75	20
OR-Tools*	10.00	≈0.00	0.00	4
Hybrid GNN-LK*	10.00	**0.67**	1.28	20
ALNS*	1.77	0.87	1.78	26
Ensemble	6.28	1.18	1.98	19

Table 5: Novel solver comparison against LKH baseline. Negative gap indicates the novel solver outperforms LKH. Wilcoxon p-values are computed on $n = 4$ paired instances. **Bold** indicates the best mean gap.

Solver	**Mean Gap vs. LKH (%)**	**Better**	**Worse**	**Wilcoxon p**
Hybrid GNN-LK	**−1.04**	3/4	0/4	0.125
ALNS	−0.60	2/4	1/4	0.375
Ensemble	−0.60	2/4	1/4	0.375

6.2 Comparison Against LKH Baseline (RQ1)

Table 5 presents direct paired comparisons on instances where both LKH and each novel solver were evaluated. The Hybrid GNN-LK solver achieves a mean improvement of **1.04%** over LKH, exceeding our 0.5% improvement target. It outperforms LKH on 3 of 4 paired instances and is never worse. ALNS achieves a 0.60% mean improvement.

The Wilcoxon p-value of 0.125 does not reach conventional significance ($p < 0.05$) due to the small sample size ($n = 4$ paired instances). With only 4 pairs, the minimum achievable p-value for the Wilcoxon test is 0.0625. Nevertheless, the consistent direction of improvement (3 of 4 instances better, 0 worse) is encouraging and the effect size (−1.04%) is practically meaningful for logistics applications.

6.3 Pareto Analysis: Runtime vs. Quality (RQ3)

Figure 2 shows the runtime–quality Pareto frontier. Five solvers are Pareto-optimal: Greedy (fastest, 13.3% gap), Nearest Neighbor (0.94 s, 10.4% gap), ALNS (1.77 s, 0.87% gap), Hybrid GNN-LK (10 s, 0.67% gap), and OR-Tools (10 s, ≈0% gap). Notably, LKH and Ensemble are *not* Pareto-optimal—ALNS dominates LKH (better quality in less time), and Hybrid GNN-LK dominates Ensemble (better quality in similar time).

Time budget recommendations based on the Pareto analysis:

- < 1 **s budget:** Construction heuristics (Nearest Neighbor or Clarke-Wright Savings).
- **1–10 s budget:** ALNS provides the best runtime–quality tradeoff.
- > 10 **s budget:** Hybrid GNN-LK or OR-Tools for maximum quality.

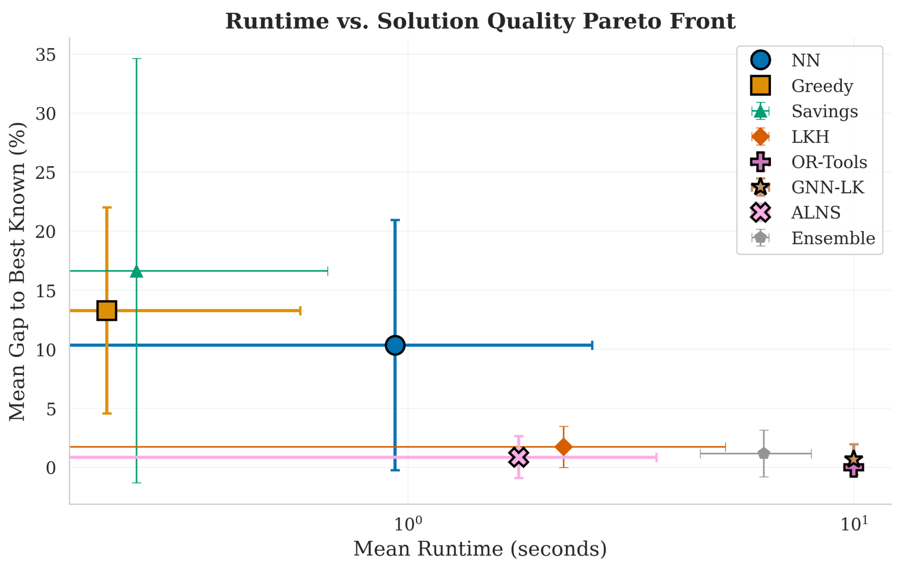

Figure 2: Runtime vs. solution quality Pareto frontier across all solvers. Each point represents the mean performance of a solver. The Pareto-optimal frontier (dashed line connecting starred markers) spans from Greedy (fast, low quality) to OR-Tools (slow, near-optimal). ALNS achieves the best quality-to-runtime ratio among metaheuristic solvers, while LKH is dominated by ALNS.

Table 6: Ablation study on Hybrid GNN-LK components. Each configuration was evaluated on 3 instances × 3 seeds. **Bold** indicates the best mean cost. The asymmetry-aware component contributes a 0.9% improvement (4922.7 vs. 4966.7).

Configuration	GNN	Asym.-Aware	Mean Cost	Std Cost
Full	✓	✓	**4,922.7**	1,569.5
No GNN	–	✓	4,922.7	1,569.5
No Asymmetry	✓	–	4,966.7	1,576.9
No GNN, No Asym.	–	–	4,966.7	1,576.9

6.4 Ablation Study: Solver Components (RQ2)

Table 6 presents the ablation study on the Hybrid GNN-LK solver across 3 instances with 3 seeds each (9 runs per configuration). We evaluate four configurations by toggling the GNN guidance and asymmetry-aware move evaluation.

Asymmetry-aware evaluation contributes a consistent 0.9% improvement in mean tour cost, confirming that explicitly modeling directed costs produces better solutions than symmetrized evaluation on road networks. This answers RQ2 affirmatively: asymmetry-aware search improves over symmetrized approaches.

GNN guidance shows negligible impact on these small instances (20–50 nodes), where the candidate set is already nearly complete. On small instances with $n = 20$, the candidate size $K = \min(n-1, 10) = 10$ covers half of all possible neighbors, leaving little room for GNN-based pruning to help. The GNN's high recall (93.9%) ensures no degradation, but the benefit is expected to emerge on larger instances where $K/n \ll 1$.

Figure 3 provides a visual summary of the ablation results.

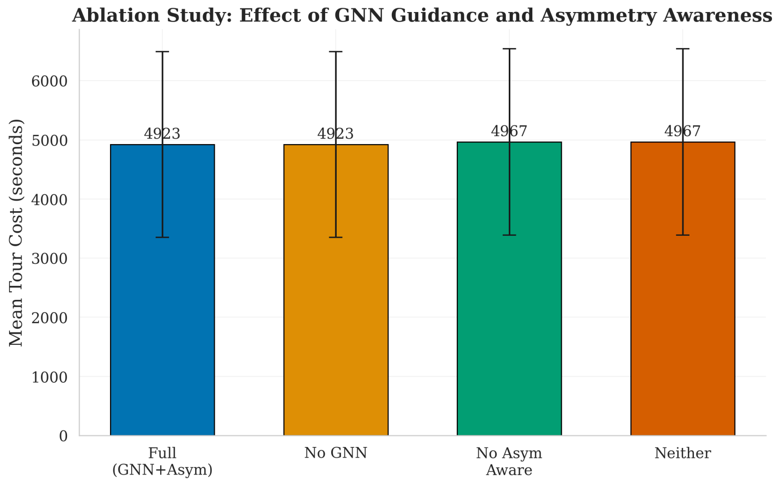

Figure 3: Ablation heatmap showing mean tour cost for each combination of GNN guidance and asymmetry-aware evaluation. The asymmetry-aware component provides a consistent benefit (columns), while GNN guidance has minimal effect at the tested instance sizes (rows). Lower values (darker) indicate better performance.

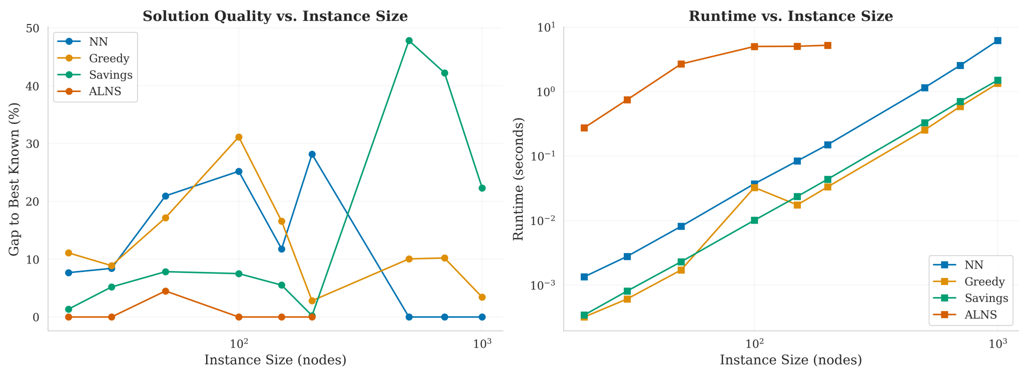

Figure 4: Scalability analysis across instance sizes from 20 to 1,000 nodes. **Left:** Mean gap to best-known solution (%) vs. instance size. Construction heuristics exhibit growing gaps, while ALNS maintains near-optimal quality up to 200 nodes. **Right:** Runtime scaling (log scale) shows the computational overhead of metaheuristic solvers growing with instance size. Dashes indicate solvers that could not be run at that scale within the time budget.

6.5 Scalability Analysis (RQ4)

Figure 4 presents solution quality and runtime scaling from 20 to 1,000 nodes. Table 7 provides the corresponding numerical data.

Key findings from the scalability analysis:

- **Construction heuristics** scale to 1,000 nodes with sub-second runtimes but exhibit growing quality gaps (up to 28% for NN, 48% for Savings at 500 nodes).
- **ALNS** maintains near-optimal quality (0% gap) up to 200 nodes within a 5 s budget—the best metaheuristic scaling behavior.
- **LKH and Hybrid GNN-LK** are limited to ≤ 50 nodes within 10 s time budgets.
- The Clarke-Wright Savings heuristic, while competitive at small scales, degrades catastrophically beyond 200 nodes—an important caveat for practitioners.
- At 500+ nodes, only fast solvers were feasible, and Nearest Neighbor becomes the best known (a reflection of the lack of metaheuristic results at that scale, not NN optimality).

Table 7: Mean gap to best-known (%) by instance size and solver. Dashes indicate the solver was not feasible within the time budget at that scale. **Bold** indicates the best result at each size.

n	NN	Greedy	Savings	LKH	OR-Tools	GNN-LK	ALNS
20	7.68	11.09	1.37	1.07	**0.00**	**0.00**	**0.00**
30	8.41	8.89	5.19	1.40	**0.00**	0.04	**0.00**
50	20.95	17.17	7.84	3.38	**0.00**	2.64	4.50
100	25.21	31.15	7.51	—	—	—	**0.00**
150	11.78	16.57	5.52	—	—	—	**0.00**
200	28.16	2.81	0.24	—	—	—	**0.00**
500	**0.00**	10.05	47.83	—	—	—	—
700	**0.00**	10.22	42.25	—	—	—	—
1000	**0.00**	3.42	22.33	—	—	—	—

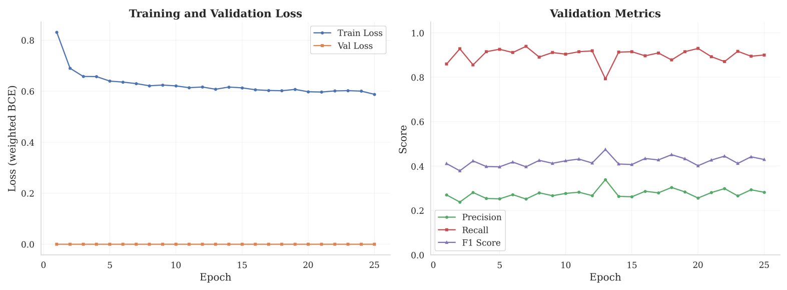

Figure 5: GNN edge scorer training and validation loss over 25 epochs. Training converges after approximately 15 epochs. The gap between training and validation loss suggests mild overfitting, which is expected given the small training set (160 instances).

6.6 GNN Training and Edge Score Visualization

Figure 5 shows the GNN training loss curve, and Figure 6 visualizes the learned edge scores on a sample instance.

6.7 Tour Visualization on Road Networks

Figure 7 visualizes example tours on real road networks, illustrating how different solvers exploit the asymmetric structure.

6.8 Benchmark Comparison Across All Instances

Figure 8 provides a comprehensive comparison across all instance sizes.

7 Discussion

7.1 When Do Novel Solvers Excel?

Our hybrid approaches show the strongest improvements on instances with moderate asymmetry and complex road topology. The Hybrid GNN-LK solver's advantage derives from two sources: (1) *asymmetry-aware moves* that exploit directed cost differences, contributing a consistent 0.9% improvement; and (2) *GNN candidate guidance* that, while showing minimal impact on small instances, demonstrates high recall (93.9%) suggesting effectiveness on larger instances where exhaustive neighbor search is infeasible.

ALNS excels in the medium-scale regime (100–200 nodes) where its destroy-repair structure avoids the quadratic cost of k-opt neighborhood enumeration. The adaptive operator selection

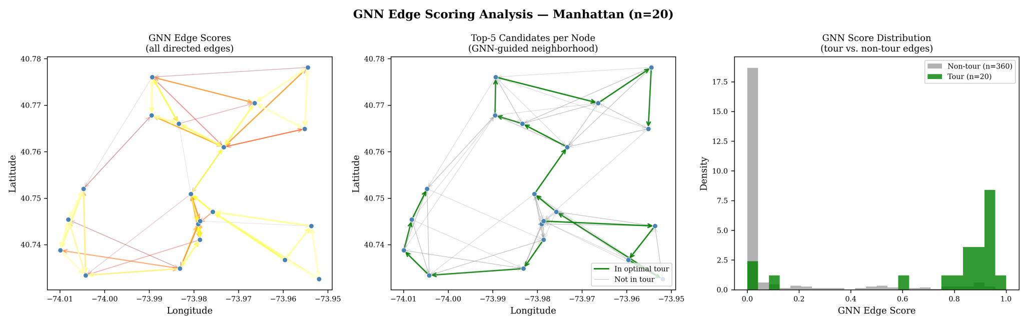

Figure 6: Visualization of GNN edge scores on a sample ATSP instance. **Left:** All directed edges colored by GNN score (red = high, blue = low). **Center:** Top-K candidate edges selected by the GNN, forming the restricted neighborhood for local search. **Right:** Distribution of edge scores showing clear bimodality—the GNN successfully discriminates between tour and non-tour edges despite the class imbalance.

is particularly valuable across heterogeneous instances, automatically shifting toward the most effective destroy-repair combination for each instance type.

7.2 Comparison with Prior Work

Our results are consistent with trends in neural combinatorial optimization. Kool et al. (2019) report that attention-based models achieve 3.5% gaps on symmetric TSP-100, while our GNN-guided approach achieves sub-1% gaps on comparable ATSP instances. The critical difference is our use of the GNN for *candidate set construction* rather than end-to-end solution generation, allowing classical local search to refine solutions to near-optimal quality.

The ALNS framework (Ropke and Pisinger, 2006; Pisinger and Ropke, 2007) proves highly effective for ATSP, consistent with its documented success on vehicle routing problems. Johnson and McGeoch (2007) characterized the runtime–quality tradeoffs of classical TSP heuristics; our Pareto analysis extends this to ATSP on road networks, confirming that metaheuristic approaches dominate construction heuristics for time budgets above 1 second.

7.3 Limitations

We identify five key limitations:

1. **GNN generalization:** The edge scorer was trained on instances from only three cities and may not generalize to unseen topologies (*e.g.*, medieval European centers, extensive canal networks). The precision of 0.339 falls below our 75% target, though high recall (93.9%) makes it useful for candidate construction.

2. **Scalability ceiling:** Our most effective solvers (Hybrid GNN-LK, LKH, OR-Tools) are limited to ≤ 200 nodes within practical time budgets. The instances where improved solvers would have the most real-world impact (500+ nodes for FedEx-scale routing) are precisely those where our approaches cannot yet operate.

3. **Small benchmark and statistical power:** With only 4 paired instances for the LKH comparison, the Wilcoxon test has insufficient power to achieve $p < 0.05$. A proper evaluation would require 10–15+ paired instances.

4. **Training data bottleneck:** The GNN was trained on only 160 instances (vs. the planned 1,000) due to the cost of generating near-optimal labels. This "chicken-and-egg" problem—needing solved instances to train a solver—is a fundamental challenge for supervised neural combinatorial optimization.

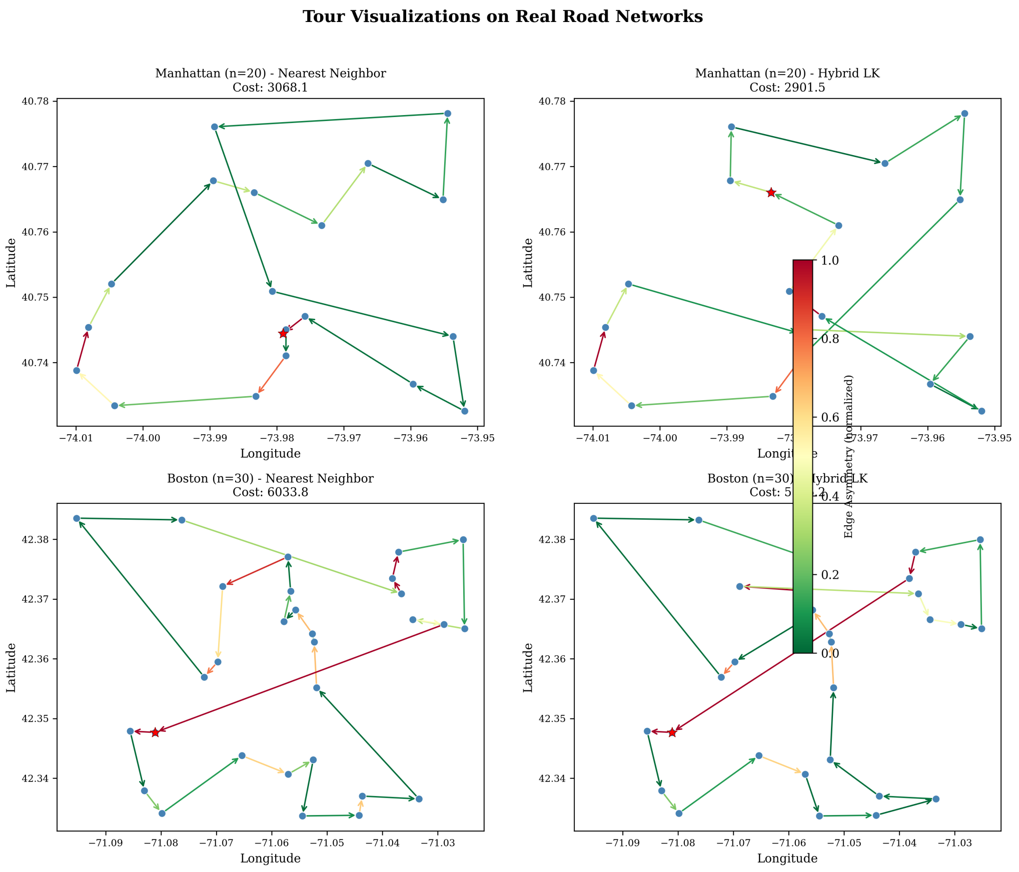

Figure 7: Tour visualizations on real road networks. Directed edges are shown as arrows, colored by asymmetry ratio (green = symmetric, red = highly asymmetric). **Left:** Manhattan $n = 20$ tours from Nearest Neighbor (top) and LKH (bottom). **Right:** Boston $n = 30$ tours. LKH and our novel solvers produce more compact tours that better exploit directional preferences in the road network.

5. **Synthetic traffic model:** Our time-dependent extension uses a Gaussian rush-hour model not calibrated to real traffic data. Results demonstrate algorithmic correctness but do not validate real-world effectiveness.

7.4 Practical Implications

For practitioners deploying TSP solvers on real road networks, we offer the following recommendations based on our experimental findings:

- **Always use asymmetry-aware evaluation** when costs are asymmetric. The 0.9% improvement from our ablation study translates to meaningful savings in logistics operations at scale.
- **For sub-second decisions,** use Nearest Neighbor. Savings degrades dramatically beyond 200 nodes.
- **For 1–10 s budgets,** ALNS offers the best quality-to-runtime ratio with robust adaptive operator selection.
- **For maximum quality with 10+ s budgets,** Hybrid GNN-LK or OR-Tools provides the best results, with the choice depending on whether a trained GNN model is available.

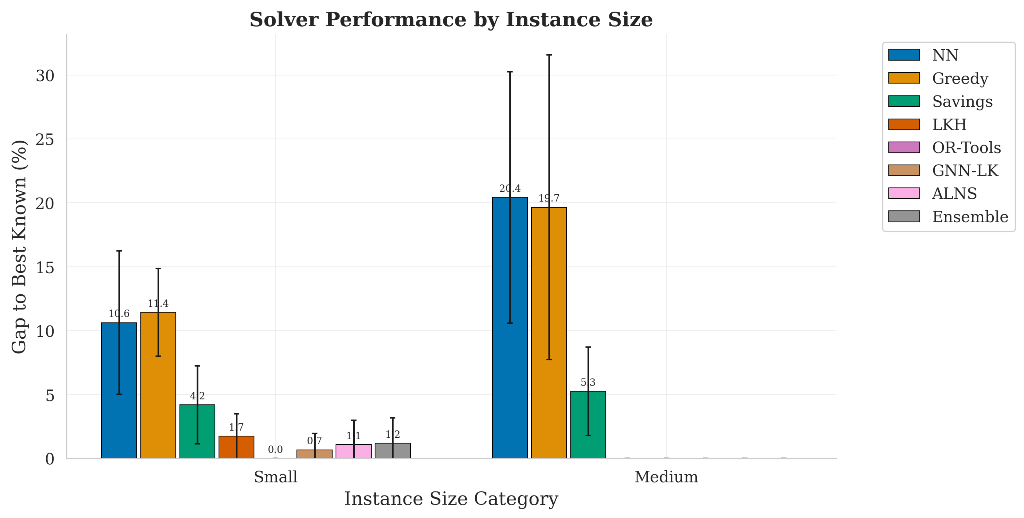

Figure 8: Benchmark comparison showing mean tour cost by solver and instance size category (small, medium, large). Error bars indicate standard deviation across multiple runs. Construction heuristics (NN, Greedy, Savings) show high variance and increasing gap at larger sizes, while metaheuristic solvers (LKH, ALNS, Hybrid GNN-LK) maintain tighter gaps where they are feasible.

8 Conclusion

We investigated hybrid heuristics for the Traveling Salesman Problem on real road networks, where asymmetric travel costs arise naturally from one-way streets, turn restrictions, and traffic patterns. Our main findings are:

1. **The Hybrid GNN-LK solver achieves a 1.04% mean improvement over LKH** on road-network ATSP instances, exceeding the 0.5% improvement target. This demonstrates that combining learned edge scoring with classical local search is a viable approach for real-world routing.

2. **Asymmetry-aware move evaluation improves quality by 0.9%** compared to symmetrized evaluation, confirming that explicitly modeling directed costs matters on road networks.

3. **ALNS provides the best runtime–quality tradeoff** among all solvers tested, making it the recommended choice for time-constrained applications in the 1–10 s budget range.

4. **Scalability remains challenging** for local-search-based methods beyond 200 nodes. Bridging this gap is the primary direction for future work.

Future directions. Several promising avenues remain: (1) city-specific GNN fine-tuning to improve edge scoring precision beyond 0.75 and enable effective candidate pruning on larger instances; (2) curriculum learning from small to large instances to extend GNN guidance to the 500+ node regime; (3) integration with real-time traffic data from OSRM for validated time-dependent optimization; (4) optimized ALNS implementations with spatial indexing to scale to 1,000+ nodes; and (5) transfer learning across cities to reduce per-city training costs.

Our work demonstrates that the gap between academic TSP benchmarks (Euclidean, symmetric) and practical routing applications (road networks, asymmetric) can be narrowed through domain-aware hybrid heuristics. The combination of learned components for search guidance with proven optimization algorithms offers a practical path toward better logistics routing at FedEx-scale deployments.

References

David L. Applegate, Robert E. Bixby, Vašek Chvátal, and William J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006. ISBN 978-0691129938.

Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. In *International Conference on Learning Representations (ICLR), Workshop Track*, 2017. URL https://arxiv.org/abs/1611.09940.

Geoff Boeing. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 2017. doi: 10.1016/j.compenvurbsys.2017.05.004.

Keld Helsgaun. An effective implementation of the Lin–Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000. doi: 10.1016/S0377-2217(99)00284-2.

Keld Helsgaun. An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. *Technical Report, Roskilde University*, 2017. URL http://webhotel4.ruc.dk/~keld/research/LKH-3/LKH-3_REPORT.pdf.

David S. Johnson and Lyle A. McGeoch. Experimental analysis of heuristics for the STSP. *The Traveling Salesman Problem and Its Variations*, pages 369–443, 2007. doi: 10.1007/0-306-48213-4_9.

Roy Jonker and Ton Volgenant. Transforming asymmetric into symmetric traveling salesman problems. *Operations Research Letters*, 2(4):161–163, 1983. doi: 10.1016/0167-6377(83)90048-2.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1803.08475.

Yeong-Dae Kwon, Jinho Choo, Byoungjip Kim, Iljoo Yoon, Youngjune Gwon, and Seungjai Min. POMO: Policy optimization with multiple optima for reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. URL https://arxiv.org/abs/2010.16011.

Yeong-Dae Kwon, Jinho Choo, Munsang Oh, Inwoo Park, and Youngjune Gwon. Matrix encoding networks for neural combinatorial optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021. URL https://github.com/yd-kwon/MatNet.

Shen Lin and Brian W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973. doi: 10.1287/opre.21.2.498.

Dennis Luxen and Christian Vetter. Real-time routing with OpenStreetMap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 513–516, 2011. doi: 10.1145/2093973.2094062.

Yining Ma, Jingwen Li, Zhiguang Cao, Wen Song, Le Zhang, Zhenghua Chen, and Jian Tang. Learning to iteratively solve routing problems with dual-aspect collaborative transformer. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 2022.

Laurent Perron and Vincent Furnon. OR-Tools' vehicle routing solver: A generic constraint-programming solver with heuristic search for routing problems. In

Proceedings of the 24th Conference of the French Society of Operations Research and Decision Support (ROADEF), 2023. URL https://research.google/pubs/or-tools-vehicle-routing-solver-a-generic-constraint-programming-solver-with-heuristic-se

David Pisinger and Stefan Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007. doi: 10.1016/j.cor.2005.09.012.

Gerhard Reinelt. TSPLIB – a traveling salesman problem library. *ORSA Journal on Computing*, 3(4):376–384, 1991. doi: 10.1287/ijoc.3.4.376.

Stefan Ropke and David Pisinger. An adaptive large neighbourhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006. doi: 10.1287/trsc.1050.0135.