

Article

Traffic-Informed Optimization of Last-Mile Delivery Using Hybrid Heuristic Approaches

Afia Yeboah ^{1,*}, Deo Chimba ¹ and Malshe Rohit ²

¹ Department of Civil and Architectural Engineering, Tennessee state University, Nashville, TN 37209, USA

² Amazon, Inc., Seattle, WA 98109, USA

* Correspondence: ayeboah2@my.tnstate.edu; Tel.: +1-615-963-5430

Abstract

The rapid growth of e-commerce has intensified operational and sustainability challenges in urban last-mile delivery, necessitating routing methods that perform reliably under realistic traffic and spatial conditions. This study evaluates three routing algorithms, Nearest Neighbor (NN), Clarke–Wright Savings (CWS), and Ant Colony Optimization (ACO), using 1764 real-world Amazon delivery stops grouped into ten operational clusters in the Nashville metropolitan area. Travel distances and times were obtained through the Google Maps Distance Matrix API in driving mode to reflect actual road network structure and typical traffic conditions. Substantial performance differences were observed across algorithms and cluster configurations. NN achieved a strong performance in compact clusters (18.43 miles and 58.48 min in Cluster 4) but performed poorly in dispersed clusters (82.44 miles and 196.48 min in Cluster 9), reflecting high sensitivity to spatial dispersion. In contrast, CWS consistently reduced travel distance and time across clusters, achieving the shortest observed route (18.50 miles and 47.82 min in Cluster 10). Relative to ACO, CWS reduced travel distance by up to 42% (Cluster 9) and reduced travel time by over 45% in high-dispersion clusters. ACO exhibited the highest variability, with distances reaching 98.77 miles and travel times exceeding 218 min. Multi-criteria evaluation using efficiency ratios, distributional analysis, performance quadrant visualization, and a Composite Performance Index (CPI) confirmed the dominance of CWS. CPI scores of 1.00 (CWS), 0.78 (NN), and 0.00 (ACO) reflected balanced spatial and temporal efficiency under identical traffic-informed inputs. The results demonstrate that deterministic savings-based routing provides superior stability, efficiency, and scalability in semi-static urban delivery systems. However, the present study did not benchmark the evaluated algorithms against state-of-the-art exact TSP solvers (e.g., Concorde, LKH) or more recent metaheuristics such as Genetic Algorithms or Variable Neighborhood Search. The objective was to provide a controlled empirical comparison under consistent traffic-informed cost matrices rather than to establish global optimality bounds. Consequently, while the findings strongly support the relative superiority of the Clarke–Wright Savings approach within the evaluated framework, future research incorporating advanced exact and hybrid optimization methods would further contextualize algorithmic performance.

Keywords: traveling salesman problem; nearest neighbor; ant colony optimization; Clarke–Wright Savings; last-mile delivery

Academic Editor: Tianwen Zhang

Received: 10 February 2026

Revised: 23 February 2026

Accepted: 25 February 2026

Published: 27 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The rapid growth of e-commerce has significantly increased consumer demand for efficient, cost-effective, and timely delivery services, especially in urban areas. The last-mile delivery stage, defined as the final segment in the logistics chain where goods are transported from a distribution center or local facility directly to the consumer, has become one of the most critical challenges facing logistics and e-commerce companies today [1]. This critical stage can account for up to 53% of the total shipment cost, and inefficiencies during this phase directly impact customer satisfaction, operational costs, and environmental sustainability [2]. The last-mile delivery problem closely aligns with the classical Traveling Salesman Problem (TSP), a well-known combinatorial optimization issue extensively studied in operations research. The TSP involves determining the shortest possible route that visits a given set of locations exactly once and returns to the original point [3]. This optimization challenge directly mirrors the practical realities of delivery routes faced by companies such as Amazon, where packages must be efficiently distributed across a vast number of locations within stringent timeframes.

Given its complexity, the TSP is classified as a nondeterministic polynomial hard problem, indicating that there are no known algorithms capable of solving all instances of TSP efficiently (within polynomial time) [4]. Consequently, researchers and industry professionals have devoted significant efforts to developing heuristic and metaheuristic algorithms capable of generating near-optimal solutions within reasonable computational timeframes. This research will specifically evaluate the performance of three prominent optimization algorithms applied to the TSP within the context of last-mile delivery. These algorithms include the Nearest Neighbor heuristic, the Clarke–Wright Savings Algorithm, and Ant Colony Optimization (ACO). Each algorithm offers unique advantages and limitations in handling large datasets and complex logistical scenarios. The Nearest Neighbor (NN) algorithm is one of the simplest and fastest heuristic methods available, which selects the nearest unvisited node sequentially; however, it does not guarantee optimal solutions and is susceptible to suboptimal local minima [5]. The Clarke–Wright Savings Algorithm represents a classical heuristic method widely used for route optimization due to its simplicity and speed, especially effective for smaller problem instances or situations requiring rapid solution turnaround [6]. Finally, the Ant Colony Optimization (ACO), inspired by the natural foraging behaviors of ants, is a probabilistic technique that iteratively improves solutions based on pheromone trails that guide solution construction, proving effective in dynamically changing environments [7].

In real-world delivery operations, one of the most unpredictable and impactful factors is traffic congestion. Drivers often encounter delays that cannot be anticipated at the time of departure from the distribution center, leading to missed delivery windows. These delays are typically not due to driver inefficiency but rather external variables such as highway congestion and sudden traffic incidents. Such inefficiencies not only compromise customer satisfaction but also contribute to increased fuel consumption and greenhouse gas emissions. Therefore, reducing travel time is not merely a logistical concern but also an environmental imperative. This study explores how real-time traffic data can be incorporated into route optimization strategies to help drivers meet delivery times by using an Amazon dataset for Nashville. By evaluating different optimization approaches, this research aims to identify algorithms that can best adapt to real-time conditions, reduce travel distance and time, and alleviate driver workload and fatigue. Through a comparative analysis of optimization algorithms, this study seeks to develop practical insights into improving last-mile delivery performance under real-world constraints. Given the critical importance of optimizing last-mile delivery operations to reduce operational costs, delivery delays, and environmental impacts, this study seeks to rigorously evaluate the effectiveness and operational suitability of the selected routing optimization algorithms under

realistic urban delivery conditions. The research applies and compares three widely used routing approaches—Nearest Neighbor (NN), Clarke–Wright Savings (CWS), and Ant Colony Optimization (ACO)—using real-world Amazon last-mile delivery data from the Nashville metropolitan area. Algorithm performance is assessed using traffic-informed travel distance and travel time obtained through the Google Maps Distance Matrix API in driving mode. The API-provided distances and durations incorporate the underlying road network structure and typical traffic estimates available at the time of query, enabling routing outcomes to reflect realistic network conditions. However, departure time-specific traffic parameters were not specified in the API requests; therefore, the retrieved travel times represent static or typical conditions rather than explicit peak- or off-peak congestion states. In addition, this study evaluates the robustness and consistency of each algorithm across delivery clusters with varying spatial configurations. To support balanced, decision-oriented comparisons, a multi-criteria performance evaluation framework is employed, incorporating efficiency ratios, performance quadrant analysis, and a Composite Performance Index (CPI) to synthesize spatial and temporal performance into an overall measure of routing effectiveness.

2. Literature Review

Amazon and other e-commerce delivery services operate extensively within Nashville, Tennessee, just like other major urban cities. The deliveries are impacted by heavy traffic congestion especially during peak hours. Hence optimizing package delivery is highly needed. Nashville is a rapidly expanding urban center with approximately 689,000 residents in the city proper and over 2.1 million in the metropolitan region, making it the 21st-largest U.S. city and 35th-largest metro area. The metro area has a notably young median age of about 37 years [8], and over 42% of residents aged 25 and older hold a bachelor's degree or higher, which is well above the national average [9]. With a median household income of around \$75,000, the area supports both strong consumer demand and substantial economic activity. Strategically located in Middle Tennessee, Nashville is within a day's drive of approximately 75% of the U.S. population, a distribution advantage frequently cited by logistics and data center operators [10]. Together, these demographic and geographic factors make Nashville an optimal logistics hub, especially for last-mile delivery networks such as Amazon's, where a skilled workforce, concentrated consumer base, and strategic location converge to support efficient e-commerce operations. However, the city's rapid growth has led to increased traffic congestion and delivery challenges, highlighting the need for advanced optimization methods. The Traveling Salesman Problem, a classic routing optimization model, becomes highly relevant in this context as Amazon seeks to minimize travel distances and delivery times across densely packed and demographically varied neighborhoods.

Numerous studies on optimization algorithms have been published by researchers to enhance solution accuracy, improve computational efficiency, and address complex real-world problems across various domains. The Nearest Neighbor (NN) algorithm is a classic greedy heuristic for solving the TSP. It constructs a route by always selecting the nearest unvisited node, beginning from a random or pre-defined starting point. This algorithm is known for its computational simplicity and speed, often used for generating initial solutions or in scenarios where computational time is a constraint. Despite its practicality, the NN algorithm suffers from significant drawbacks. Its greedy nature leads to suboptimal solutions, particularly when the nearest choice at each step does not contribute to the global optimum. Rosenkrantz et al. provided one of the earliest analyses, demonstrating that while NN performs adequately in small datasets, its error margin grows with increasing node size [5]. Further evaluations by Reinelt [11] and Johnson & McGeoch confirm that NN's average error ranges between 25 and 30% above the optimal

tour length in large TSP instances [12]. Additionally, the algorithm does not incorporate real-time constraints such as traffic congestion or delivery time windows, making it less suitable for dynamic logistics environments like last-mile delivery. Nevertheless, NN's computational efficiency makes it valuable in hybrid optimization systems or as a benchmark for evaluating more complex algorithms. Gendreau and Potvin also suggested its utility in route initialization for advanced metaheuristics [13]. Additionally, the Clarke–Wright Savings Algorithm is a well-established heuristic developed for solving vehicle routing problems (VRPs), which are generalizations of the TSP. It begins with individual routes from the depot to each customer and iteratively merges them by calculating 'savings' in cost when two routes are combined. Clarke and Wright originally introduced the algorithm, and it remains popular due to its speed and simplicity. It is particularly useful in distribution systems requiring quick approximations [6]. Gendreau and Hertz and Toth and Vigo have shown its effectiveness in reducing delivery costs when used with capacity and time window constraints [14,15]. However, the algorithm's reliance on pairwise savings calculations makes it sensitive to input ordering and initial conditions. It also assumes symmetrical travel costs and does not naturally support real-time decision-making. As a result, while Clarke–Wright performs well in static or lightly constrained environments, its application in dynamic last-mile delivery is limited. Nevertheless, it continues to be integrated into more complex hybrid metaheuristics, as highlighted by Tan et al. and Osman & Kelly, who demonstrate enhanced performance in real-world logistics [16,17].

Finally, Ant Colony Optimization (ACO) is a bio-inspired metaheuristic introduced by Dorigo et al. in 1996 [7] that models the pheromone-laying behavior of ants to discover the shortest paths. In TSP contexts, ACO simulates multiple agents (ants) that build routes probabilistically based on pheromone intensity and visibility (inverse of distance) [18]. After each iteration, pheromones are updated based on route quality. ACO is particularly suited to dynamic and real-time systems due to its feedback-based adaptation. A study conducted by Dorigo and Gambardella showed that ACO can outperform traditional heuristics in solution quality [19]. Further refinements by Stützle and Hoos [20] and Socha and Dorigo [21] demonstrate the algorithm's ability to handle large TSP instances with reasonable computational effort. However, parameter tuning such as evaporation rate and influence weights is crucial. Improper tuning can lead to premature convergence or excessive exploration. ACO also requires more computational resources than simpler heuristics, which can be a drawback in constrained environments. Nonetheless, its flexibility and parallelizability make it a leading candidate for adaptive logistics optimization, especially when integrated with real-time data [20,21].

While classical heuristics such as NN, CWS, and ACO provide foundational approaches for combinatorial routing problems, contemporary research increasingly integrates traffic dynamics, heterogeneous constraints, and probabilistic modeling to enhance real-world applicability. For instance, a recent study by Liu and Wang proposes a dynamic vehicle routing framework integrating Adaptive Large Neighborhood Search (ALNS) with real-time traffic flow analysis, demonstrating the performance gains of congestion-aware route adaptation over static routing models [22]. Similarly, a study by Peric et al. formulates a heterogeneous fleet vehicle routing problem incorporating traffic congestion, vehicle capacity constraints, delivery time windows, and service requirements, highlighting structural differences between constrained VRP formulations and simplified TSP-based approaches [23]. Beyond routing heuristics, transportation research has integrated wavelet-based dynamic segmentation and multi-criteria decision-making (MCDM) methods for roadway safety prioritization. The study by Shirmohammadi et al. applies continuous wavelet transform (CWT) and an Analytic Hierarchy Process (AHP) for identifying and prioritizing accident-prone segments, illustrating the integration of mathematical signal processing and structured decision frameworks in transportation systems

analysis [24]. Soft computing approaches have also been introduced in transportation engineering to manage uncertainty. For example, fuzzy logic has been applied to evaluate the resilient modulus performance of asphalt mixtures under uncertain conditions, demonstrating the applicability of rule-based inference systems in infrastructure performance modeling [25]. In addition, probabilistic modeling techniques such as conjugate Bayesian analysis have been developed for performance measurement of two-lane roads under vehicular platooning conditions. A sustainability study formulates Bayesian performance models to estimate operational metrics including percent time spent following (PTSF), average travel speed, and level of service (LOS) under varying platoon penetration rates. The results demonstrate that increasing platooning intensity significantly alters traffic flow characteristics and degrades operational performance unless passing opportunities and roadway geometry are appropriately adjusted. By explicitly modeling parameter uncertainty through posterior distributions, the framework provides probabilistic confidence intervals rather than deterministic point estimates, thereby enhancing robustness in transportation system evaluation [26].

Collectively, these contributions demonstrate the expanding methodological landscape in transportation optimization, including dynamic routing, heterogeneous VRP modeling, uncertainty-aware performance evaluation, and multi-criteria decision support systems. In contrast, the present study isolates classical heuristic and metaheuristic routing algorithms under consistent spatial and traffic-informed inputs, providing a controlled benchmarking framework for comparative algorithm evaluation using real-world delivery data.

The demographic growth, mixed land-use development, and spatial heterogeneity of Nashville provide a relevant testbed for evaluating routing algorithms with differing structural properties. The metropolitan area includes a dense urban core (e.g., midtown and downtown high-rise districts) alongside lower-density suburban neighborhoods, creating both compact and spatially dispersed delivery clusters. This spatial variability makes it appropriate to evaluate algorithms that perform differently under varying network structures. The Nearest Neighbor (NN) heuristic provides a computationally simple greedy baseline suitable for compact clusters but sensitive to dispersion. The Clarke–Wright Savings (CWS) algorithm aligns with depot-based distribution systems typical of last-mile delivery operations and is designed to efficiently merge routes under centralized dispatch structures. Ant Colony Optimization (ACO), as a probabilistic metaheuristic, is particularly relevant in complex routing environments where traffic variability and network topology may influence solution exploration. Evaluating these three algorithm classes within Nashville’s heterogeneous urban structure enables a balanced comparison across deterministic heuristic and adaptive metaheuristic approaches under realistic delivery conditions.

Despite significant advances in dynamic vehicle routing, heterogeneous VRP modeling, and uncertainty-aware transportation analysis, comparatively limited research provides systematic empirical benchmarking of foundational routing heuristics using consistent real-world, traffic-informed cost matrices derived from operational last-mile delivery datasets. Many comparative studies rely on synthetic benchmark instances or varying cost assumptions, which can obscure direct performance comparison under realistic spatial and congestion conditions. In particular, empirical evaluations using actual commercial delivery data remain relatively scarce due to data accessibility constraints. The present study addresses this need by evaluating Nearest Neighbor, Clarke–Wright Savings, and Ant Colony Optimization under identical API-derived travel distance and time inputs using a real-world Amazon last-mile delivery dataset, thereby establishing a transparent and reproducible baseline for heuristic performance in practical urban delivery contexts.

3. Methodologies and Data Used

Several tasks were undertaken to achieve the objectives of this research. These include the collection and preprocessing of real-world delivery data from Amazon, a comprehensive description of the dataset, and the application of three optimization algorithms. Additionally, the methodological framework for analyzing and comparing the performance of each algorithm is discussed in detail.

3.1. Data Used

This study utilizes a comprehensive dataset of Amazon package deliveries within the Nashville metropolitan area, comprising 1764 individual delivery records. Each record captures extensive attributes related to last-mile logistics operations, providing a robust foundation for evaluating and optimizing delivery algorithms. The dataset includes precise geospatial information with latitude and longitude coordinates for each delivery stop, supplemented by street-level address details, zip codes, and city identifiers which were used for spatial analysis and route mapping. The data also includes temporal dimensions which are recorded through multiple timestamp fields such as assignment event time, enter time, exit time, and delivery window arrival and departure times. This was used for the detailed analysis of delivery scheduling, time window adherence, and route sequencing. Each package is uniquely identified by a tracking ID and linked to its corresponding Route ID, station code, and route zone, which this paper used to facilitate clustering. The ten delivery clusters analyzed in this study were derived directly from the operational Route ID field contained in the Amazon dataset. Each Route ID corresponds to a pre-defined delivery route originating from a designated station (depot) and includes multiple associated drop-off stops assigned during dispatch planning. No additional clustering algorithm was applied. The evaluation therefore reflects algorithmic performance within operationally defined delivery routes. Prior to optimization analysis, the dataset underwent preprocessing to ensure data integrity. Records were screened for completeness of critical geospatial attributes (latitude and longitude), which are required for routing computation. The final analytical dataset consisted of 1764 delivery stops with complete spatial coordinates. Duplicate tracking IDs were not detected. Missing values were minimal and occurred only in non-critical descriptive fields; these were excluded from routing calculations. In addition to locational and temporal data, the dataset documents package-level physical characteristics, including package width, height, and depth (in centimeters), along with executor capacity and planning minutes, which can be used for modeling vehicle loading constraints and delivery duration estimation. The data also includes variables such as stop type (e.g., station, drop-off), stop number, wave, and group that provide further context for delivery route structure and batching strategies. The dataset also includes calculated measures like previous-to-current meters, representing inter-stop travel distances, which are critical for assessing routing efficiency and travel time predictions.

These 1764 delivery records from an Amazon dataset for Nashville were classified into ten (10) route clusters through the lens of the TSP, and the transportation logistics provide insights into stop durations, delivery patterns by zip code, and operational inefficiencies. Figures 1 and 2 show the distribution of package delivery stops by zip code. As shown, most of the package deliveries analyzed in this study are concentrated within four key zip code areas in Nashville, 37208, 37209, 37218, and 37203, which together account for over 95% of all stops in the dataset. Zip code 37208 primarily covers the historically urban North Nashville area, including neighborhoods like Germantown and Buchanan Arts District. This region has experienced rapid residential redevelopment and densification, resulting in a high population density of urban dwellings and multi-family housing that drives significant e-commerce delivery demand. Zip code 37209 encompasses the West Nashville and Sylvan Park area, which includes a mix of older residential

neighborhoods and new infill developments. The zone is notable for its mix of single-family homes, town homes, and small businesses, all of which contribute to steady package delivery volumes. Zip code 37218 covers the Bordeaux and Northwest Nashville area, which is predominantly residential but includes several distribution routes connecting to suburban edges. This area is characterized by single-family homes on larger lots, but increased online shopping trends have contributed to notable last-mile delivery activity. Zip code 37203 includes Midtown, the Gulch, and parts of downtown Nashville, a dense urban core with numerous apartment complexes, condos, and commercial mixed-use developments. Its concentration of residential high-rises and businesses explains the continuing high frequency of package deliveries. Figure 1 shows a map of the delivery stops/location across Nashville. The data records one starting point for each route, called a “Station/Depot,” and multiple “Dropoff” points that come after it. The stop number column shows the order in which the points are delivered. A visual plot is shown in Figure 2 showing the zip codes by the number of delivery stops and highlighting significant delivery activity concentrated in specific areas such as 37209, 37208, and 37203.

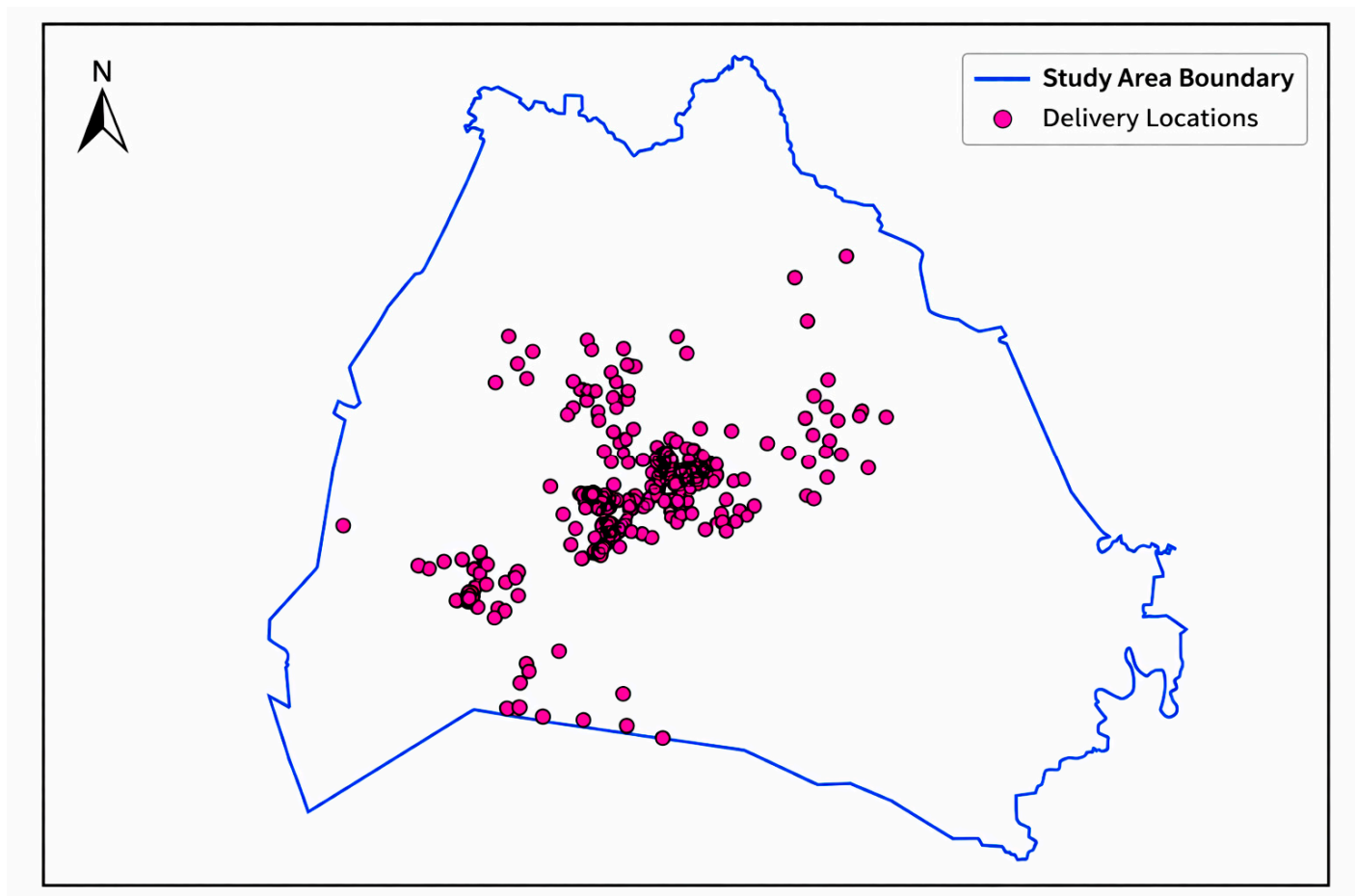


Figure 1. Delivery stops across Nashville.

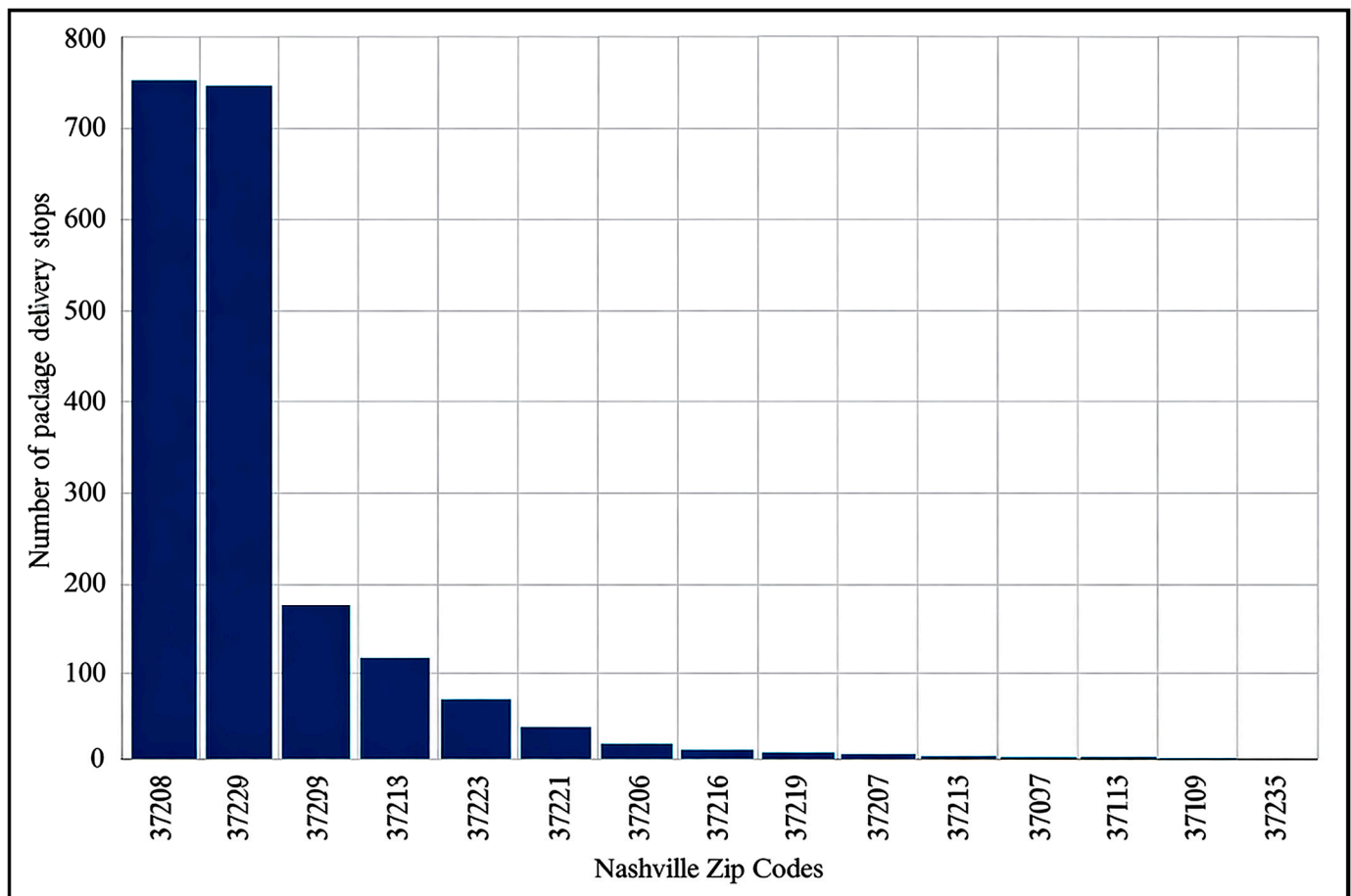


Figure 2. Top 15 zip codes by number of delivery stops.

3.2. Optimization Methods

Several optimization techniques were applied to the Amazon last-mile delivery dataset to evaluate their effectiveness in minimizing travel distance and travel time under realistic traffic conditions. Each algorithm was evaluated as a standalone routing strategy to establish baseline performance prior to hybridization or real-time deployment. This approach enables a controlled comparison of algorithmic strengths and limitations under identical spatial and traffic-informed conditions, providing a clear foundation for subsequent multi-criteria performance evaluation.

The routing problems were formulated as incapacitated single-vehicle routing instances (TSP-style problems) within each cluster. Operational constraints such as vehicle capacity limits, delivery time windows, driver rest requirements, and service times at stops were not incorporated into the optimization model. This modeling scope allows for a controlled comparison of algorithmic efficiency based solely on travel distance and travel time under identical spatial and traffic-informed conditions.

Equations (1)–(12) presented in this study correspond to established formulations of the Nearest Neighbor heuristic, Clarke–Wright Savings Algorithm, and Ant Colony Optimization as documented in the vehicle routing literature. These equations were not derived from field data nor were they introduced as novel algorithmic contributions. Instead, the real-world delivery dataset and traffic-informed travel metrics serve as inputs to these established routing frameworks. Relevant references are cited within each algorithm subsection to ensure transparency and reproducibility.

3.3. Nearest Neighbor (NN)

Nearest Neighbor is a greedy heuristic that aims to build a simple, efficient routing approach by visiting the nearest unvisited delivery point at each step. This approach begins at a specific node (often the depot) and travels to the nearest unvisited point until all have been visited, resulting in a complete tour. The process can be described as follows:

The initialization of the algorithm shown in Equation (1) starts by selecting an arbitrary point as the starting point of the tour, typically the depot denoted as x_1 . S represents the set of all delivery points excluding the depot.

$$S = \{1, 2, \dots, N\} \setminus \{x_1\} \quad (1)$$

This step ensures that the remaining unvisited points are tracked throughout the algorithm. At each iteration k , the algorithm selects the nearest unvisited point x_{k+1} from the current point x_k . This is done using the decision rule in Equation (2) where $d(x_k, j)$ is the distance between the current point x_k and a candidate point j in the unvisited set S . After selecting x_{k+1} , it is removed from S , ensuring that no point is visited more than once.

$$x_{k+1} = \arg \min_{j \in S} d(x_k, j) \quad (2)$$

The process is repeated iteratively. At each step, the closest point from the current location is added to the tour, and the set S is updated by removing the newly visited point. This continues until all points have been included in the route. After visiting all points, the algorithm completes the tour by returning to the depot (starting point), x_1 . The cost of the final connection is $d(x_N, x_1)$ where x_N is the last point visited before returning to the depot. The objective of the NN algorithm is to minimize the total distance (or time) required to complete the tour. The total tour distance is computed as shown in Equation (3). Equation (3) sums up the distance between consecutive points in the constructed route including the final leg that returns to the depot.

$$(x) = \sum_{k=1}^{N-1} d(x_k, x_{k+1}) + d(x_N, x_1) \quad (3)$$

The Nearest Neighbor construction heuristic is widely described in the contemporary vehicle routing literature as a baseline greedy route construction method [27].

3.4. Clarke–Wright Savings (CWS)

The Clarke–Wright Savings Algorithm is a constructive heuristic that aims to build efficient delivery routes by identifying cost-saving opportunities between delivery points. Initially, each customer is assigned a separate route, which involves each vehicle leaving the depot, visiting a single customer, and then returning. For customer i , the initial route distance is $d_{oi} + d_{i0}$. The total distance for such individual routes is shown in Equation (4) where d_{oi} is the distance from the depot (point 0) to customer i and N is the total number of customers.

$$D_{\text{initial}} = \sum_{i=1}^N (d_{oi} + d_{i0}) = 2 \sum_{i=1}^N d_{oi} \quad (4)$$

For each pair of customers i and j , the potential savings from merging their individual routes into a single route (Depot $\rightarrow i \rightarrow j \rightarrow$ Depot) is computed using the savings formula in Equation (5) where d_{ij} is the direct distance between customer i and j . The savings value S_{ij} represents the distance saved by connecting customers directly instead of routing each through the depot.

$$S_{ij} = d_{oi} + d_{oj} - d_{ij} \quad (5)$$

The computed savings values S_{ij} for all possible customer pairs are sorted in descending order. This ensures that the most cost-effective (highest saving) route merges are considered first during the next step. Starting from the highest savings value, customers i and j are merged into a single route only if neither customer has been previously merged

into another route that would cause duplication and if the merge maintains a valid single connected tour structure without creating subtours. In this study, the Clarke–Wright procedure was implemented in an incapacitated form consistent with the available dataset. Since no stop-level demand quantities or vehicle capacity parameters were provided, no capacity–feasibility constraints were applied during route merging.

Once all feasible merges have been completed, the overall distance of the optimized routes is determined in Equation (6) where $\sum S_{ij}^{(used)}$ is the sum of the savings values for all routes that were actually used.

$$D_{\text{final}} = D_{\text{initial}} - \sum S_{ij}^{(used)} \quad (6)$$

The savings calculation and route-merging logic follow the established Clarke–Wright framework as documented in modern vehicle routing applications and expositions [28,29].

3.5. Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is a nature-inspired metaheuristic algorithm that mimics the foraging behavior of actual ants. In the context of the TSP, artificial ants build tours probabilistically using both learned experiences (pheromones) and problem-specific heuristics (for example, distance). Better solutions arise over time as successful routes are reinforced with pheromones. The algorithm begins by initializing the pheromone trail between each pair of points (i, j) to a small positive constant $\tau_{ij}(0)$. A visibility measure η_{ij} is defined to express the heuristic desirability of moving from point i to j , typically based on the inverse of the distance in Equation (7) where d_{ij} is the distance between point i and point j .

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (7)$$

Each ant creates a complete tour by going from point to point according to a probabilistic decision process. The probability $p_{ij}^{(k)}$ of ant k migrating from point i to an unvisited point j is determined by Equation (8) where α controls the influence of the pheromone trail, β controls the influence of visibility. Each ant repeats this process until a full tour is completed.

$$p_{ij}^{(k)} = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in \text{allowed}_k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} \quad (8)$$

After all ants complete their tours, the total distance L_k for each and k is determined. The pheromone trails are then updated using two processes, evaporation (Equation (9)) and deposition (Equation (10)), where $\rho \in (0, 1)$ is the pheromone evaporation rate, Q is a constant representing pheromone quantity, m is number of ants, and L_k is the tour length for ant k .

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} \quad (9)$$

$$\Delta\tau_{ij}^{(k)} = \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ used edge } (i, j) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (11)$$

At each iteration, the best solution (shortest tour) is recorded. An elitist technique can be used, in which greater pheromones are deposited on the best-so-far tour to accelerate convergence. The process of solution construction and pheromone updating is repeated for a predefined maximum of 100 iterations. No stagnation-based convergence criterion

(e.g., no-improvement threshold) was implemented. The ACO implementation used the following parameter settings for all clusters: pheromone importance $\alpha = 1$, heuristic importance $\beta = 5$, evaporation rate $\rho = 0.5$, pheromone constant $Q = 1.0$, and number of ants $m = 10$. The maximum number of iterations was set to 100. The algorithm terminated upon reaching the predefined maximum iteration count. The algorithm's purpose is to reduce the total distance of the constructed tour. This objective function in Equation (12) evaluates and compares the ants' trips.

$$f(x) = \sum_{k=1}^{N-1} d(x_k, x_{k+1}) + d(x_N, x_1) \quad (12)$$

The transition probability rule and pheromone evaporation/deposition update follow the canonical ant system formulation introduced by [18,19].

Table 1 provides a comparative overview of the three optimization methods used to solve the TSP using Amazon delivery data. It highlights each method's model type, key strengths, computational complexity, and suitability for large-scale datasets. Table 2 summarizes the default parameter settings and implementation assumptions used for the NN, CWS, and ACO algorithms. NN was initialized at the depot and constructed greedily, CWS used the original Clarke–Wright Savings formulation without capacity constraints, and ACO parameters (α , β , ρ , ants, and iterations) were fixed to balance exploration and exploitation under a closed-tour formulation.

Table 1. Summary of algorithms.

Method	Model Type	Key Strength	Complexity	Suitable For Large Data?
Nearest Neighbor	Heuristic	Fast, simple	Low	Yes
Clarke–Wright Savings	Heuristic	Efficient, interpretable	Low	Yes
Ant Colony Optimization	Metaheuristic	Flexible, parallelizable	Medium	Yes

Table 2. Default parameter settings for implemented routing algorithms.

Algorithm	Parameter	Value
NN	Starting node selection	Depot-based
CWS	Savings formula	$s_{ij} = d_{0i} + d_{0j} - d_{ij}$
CWS	Capacity constraint	Not applied (incapacitated)
ACO	α	1
ACO	β	5
ACO	ρ (evaporation rate)	0.5
ACO	Number of ants	10
ACO	Maximum iterations	100

4. Results

This section presents the results derived from applying three distinct optimization algorithms, Ant Colony Optimization (ACO), Clarke–Wright Savings (CWS), and Nearest Neighbor (NN), to route optimization across ten delivery clusters. Each algorithm's performance was evaluated based on two key criteria: total travel distance and travel time. These metrics are essential for understanding routing efficiency, operational cost implications, and service quality in real-world logistics and transportation scenarios. The subsequent comparative analysis and sensitivity assessments further highlight the relative strengths, limitations, and practical applicability of each optimization method, facilitating informed decision-making and strategy development in transportation planning and logistics management.

4.1. Nearest Neighbor

4.1.1. Route Metrics and Efficiency

The analysis revealed performance variability across the clusters, primarily influenced by the spatial distribution of delivery points. The key metrics for each cluster's optimized route are summarized below in Table 3.

Table 3. Comparative route metrics table.

Cluster ID	NN Distance (m)	NN Travel Time (min)	CWS Distance (m)	CWS Travel Time (min)	ACO Distance (m)	ACO Travel Time (min)
1	37.57	109.38	36.56	61.48	66.49	173.52
2	26.7	81.48	37.45	68.3	53.42	140.47
3	30.68	101.72	30.29	54.32	62.57	187.9
4	18.43	58.48	26.43	46.72	45.05	159.98
5	46.4	123.37	54.99	94.73	84.49	207.25
6	24.25	74.28	34.2	51.63	47.09	125.65
7	32.48	121.5	32.96	62.93	62.17	218.4
8	22.12	80.67	28.32	55.07	44.75	148.45
9	82.44	196.48	57.21	110.52	98.77	218.25
10	45.32	145.22	18.5	47.82	58.69	174.8

4.1.2. Observations and Insights

The Nearest Neighbor (NN) algorithm exhibited notable variability in route performance across the different delivery clusters:

- **Shortest Route—Cluster 4:** This cluster produced the most efficient route, as shown in Figure 3, with a total distance of 18.43 miles and a travel time of 58.48 min. Its compact spatial configuration facilitated optimal sequencing and minimal travel.
- **Longest Route—Cluster 9:** Cluster 9 in Figure 3 resulted in the least efficient route, covering 82.44 miles in 196.48 min. The high dispersion of delivery points led to excessive backtracking, highlighting the limitations of NN's greedy approach.

The remaining clusters displayed moderate performance levels, influenced by varying densities and layouts of delivery points. For instance, Clusters 2, 3, 6, and 8 exhibited relatively efficient routing, each maintaining distances under 31 miles and travel times under 102 min, suggesting moderately compact layouts. In contrast, Clusters 1, 5, 7, and 10 experienced longer routes ranging from 32.48 to 46.40 miles and up to 145.22 min indicating less spatial cohesion and a greater impact from NN's myopic decision-making. Overall, these findings underscore the Nearest Neighbor algorithm's sensitivity to initial node selection and spatial configuration. Compact clusters generally yielded efficient results, whereas dispersed clusters significantly increased route length and delivery time.

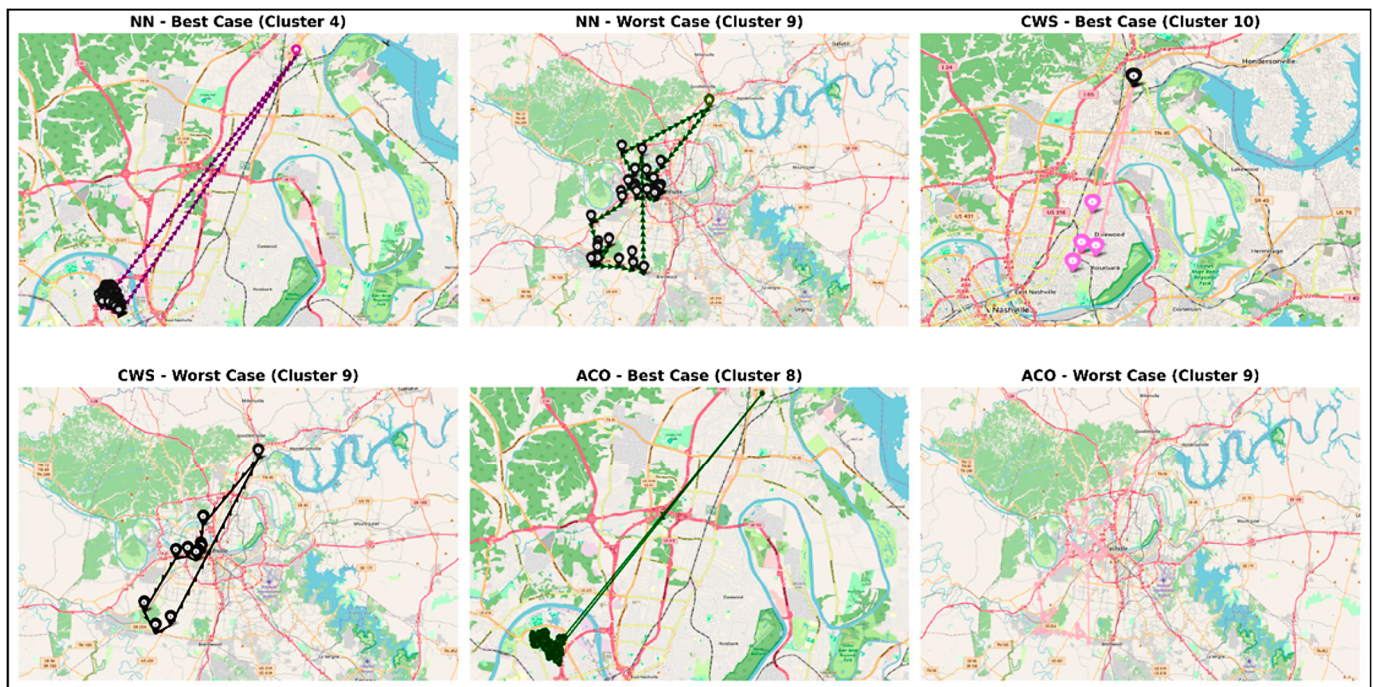


Figure 3. Best vs. worst route optimization across NN, CWS, and ACO.

4.2. Clarke–Wright Savings

4.2.1. Route Metrics and Efficiency

The Clarke–Wright Savings Algorithm was implemented to optimize delivery routes across ten independent clusters, each anchored by a designated station acting as the depot. The algorithm aimed to minimize total travel distance and time, with real-time distance and duration inputs retrieved from the Google Distance Matrix API. The optimized results for each cluster are summarized in Table 3.

4.2.2. Observations and Insights

The results indicate significant variability across clusters, reflecting differences in spatial configuration and roadway conditions.

- Cluster 9 recorded the longest route (57.21 miles, 110.52 min), likely due to sparse delivery points or network congestion, as shown in Figure 3.
- Cluster 5 also showed relatively high values, suggesting logistical challenges within suburban or complex urban zones.
- Cluster 10, with the shortest distance (18.50 miles) and low travel time (47.82 min), demonstrated the algorithm’s efficiency in dense, compact areas, as shown in Figure 3.
- Cluster 4 also achieved a strong performance, underscoring how spatial compactness supports route optimization.

The integration of Google API data enhanced the algorithm’s practicality by accounting for real-world travel conditions. While Clarke–Wright provided efficient baseline routing, the observed variability suggests potential for improvement via hybrid models that incorporate metaheuristics (e.g., Genetic Algorithm, ACO) or machine learning for adaptive decision-making.

4.2.3. Ant Colony Optimization

Route Metrics and Efficiency

This section presents the results of the Ant Colony Optimization (ACO) algorithm applied to optimize last-mile delivery across ten distinct clusters. Each cluster was serviced from a central depot and consisted of multiple delivery locations. The goal of optimization was to minimize both total travel distance and time, leveraging real-world geospatial data and dynamic routing paths. The optimized results for each cluster are summarized in Table 3.

4.2.4. Observations and Insights

The ACO algorithm demonstrated adaptive performance across varying spatial configurations.

- Cluster 8 in Figure 3 recorded one of the most efficient outcomes with only 44.75 miles and 148.45 min, suggesting an optimal match between delivery point proximity and pheromone-based decision rules.
- Cluster 6 also showed a high performance, maintaining a relatively low distance (47.09 miles) and time (125.65 min), which points to good local exploration and exploitation balance.
- On the other hand, Cluster 5 and Cluster 9 in Figure 3 exhibited the longest distances (84.49 and 98.77 miles, respectively), with travel times exceeding 207 min, indicating complex routing challenges and potentially dispersed delivery layouts.
- Cluster 7 recorded the longest travel time (218.40 min) despite a shorter distance than Cluster 9, suggesting possible inefficiencies due to traffic-prone routes or suboptimal path convergence.

A direct comparison across clusters highlights notable differences in route length and travel time among the evaluated algorithms. In Cluster 9, which represents one of the more spatially dispersed delivery configurations, the CWS algorithm generated a route distance of 57.21 miles, compared to 82.44 miles for NN and 98.77 miles for ACO. This corresponds to approximate reductions of 30% relative to NN and 42% relative to ACO. In Cluster 10, CWS produced a route of 18.50 miles, whereas NN required 45.32 miles. Travel time patterns followed a similar trend. For example, in Cluster 1, CWS completed the route in 61.48 min compared to 109.38 min for NN and 173.52 min for ACO. These contrasts illustrate measurable differences in distance and duration across varying spatial layouts prior to the multi-criteria evaluation presented in the next section.

Overall, the ACO algorithm effectively navigated various cluster scenarios, balancing exploration and exploitation principles inherent in nature-inspired metaheuristics. However, performance varied based on cluster density, delivery point distribution, and traffic variability. The findings suggest that while ACO provides a flexible framework for route optimization, its effectiveness can be enhanced by integrating real-time traffic data, dynamic pheromone decay mechanisms, or hybridization with other models such as Genetic Algorithms or Reinforcement Learning for improved global search capabilities.

4.3. Multi-Criteria Performance Evaluation and Trade-Offs

While individual distance and travel time metrics provide useful insights, practical routing decisions often require a balanced evaluation across multiple criteria. To address this, a multi-criteria performance assessment framework was adopted. To objectively examine the performance of the three routing algorithms, different visualizations were used to compare distance, travel time, and efficiency across the ten delivery clusters. Figures 4 and 5 depict the optimal route distances (in miles) and the travel times (in minutes) over ten clusters for the ACO, NN, and CWS algorithms. The CWS algorithm consistently finds

the shortest distances, displaying greater spatial efficiency through its cost-effective route merging technique. NN performs moderately but is constrained by greedy, local decision-making, resulting in poor pathways in dispersed clusters. ACO produces the longest pathways, indicating an adventurous tendency and greater solution diversity. Overall, CWS provides the most efficient distance minimization across all clusters. The CWS yielded the overall shortest travel times, confirming its effectiveness in both spatial and temporal route compression. NN performed competitively in tightly grouped clusters but lacked consistency due to its locally optimal, short-sighted path selection strategy. ACO produced the longest durations, primarily due to its iterative exploration and behavior. The results highlight CWS as the most time-efficient algorithm for last-mile delivery.

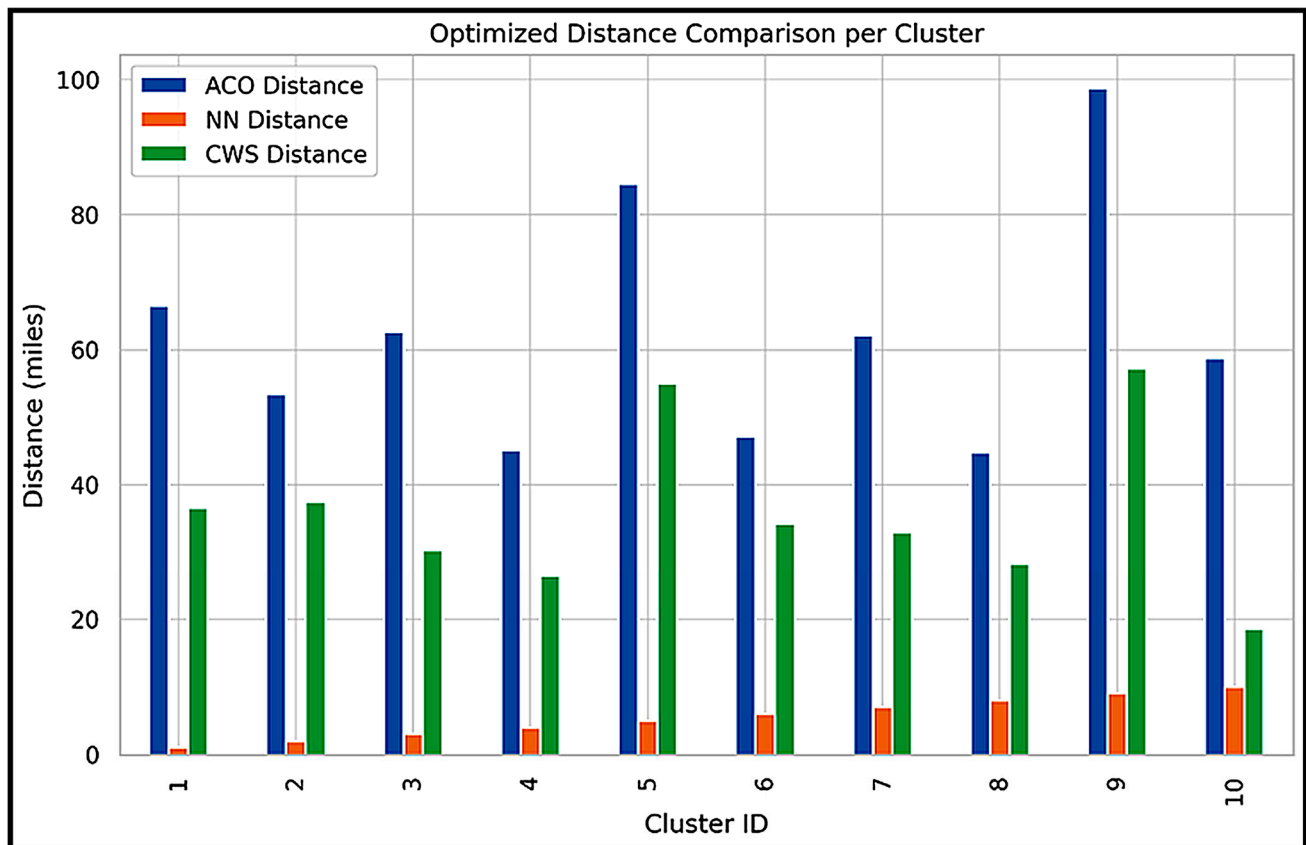


Figure 4. Comparison of total route distance (miles) across optimization algorithms.

Figure 6 presents boxplots illustrating the distribution of optimized distances and travel times for ACO, NN, and CWS across all clusters. ACO exhibits the highest variability, particularly in travel time, indicating inconsistent convergence and sensitivity to route complexity. NN shows narrow spreads in distance but broader variance in time, reflecting its unstable performance in diverse spatial layouts. CWS maintains the most compact distributions in both metrics, confirming its stability and reliability in generating consistently efficient routes. These patterns underscore CWS's advantage not just in mean performance but also in operational predictability.

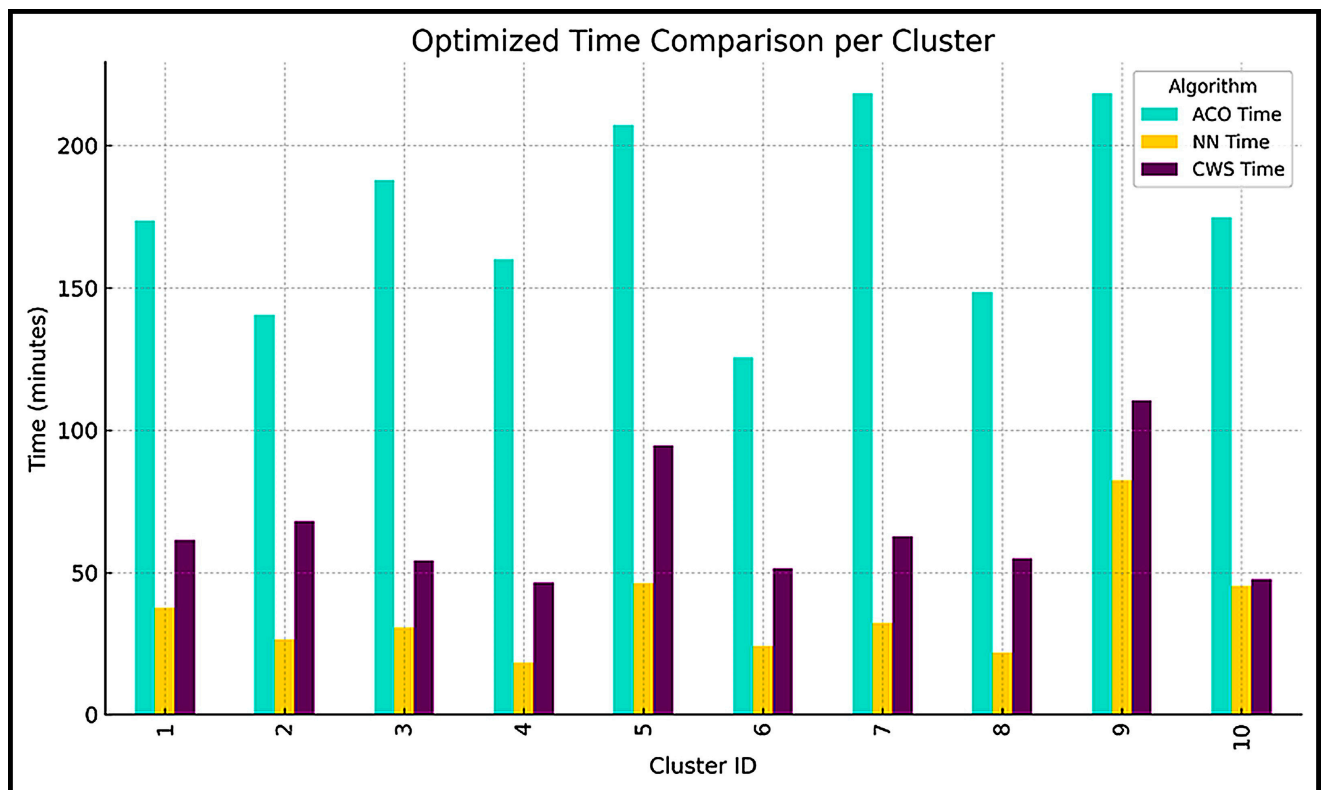


Figure 5. Comparison of total travel time (minutes) across optimization algorithms.

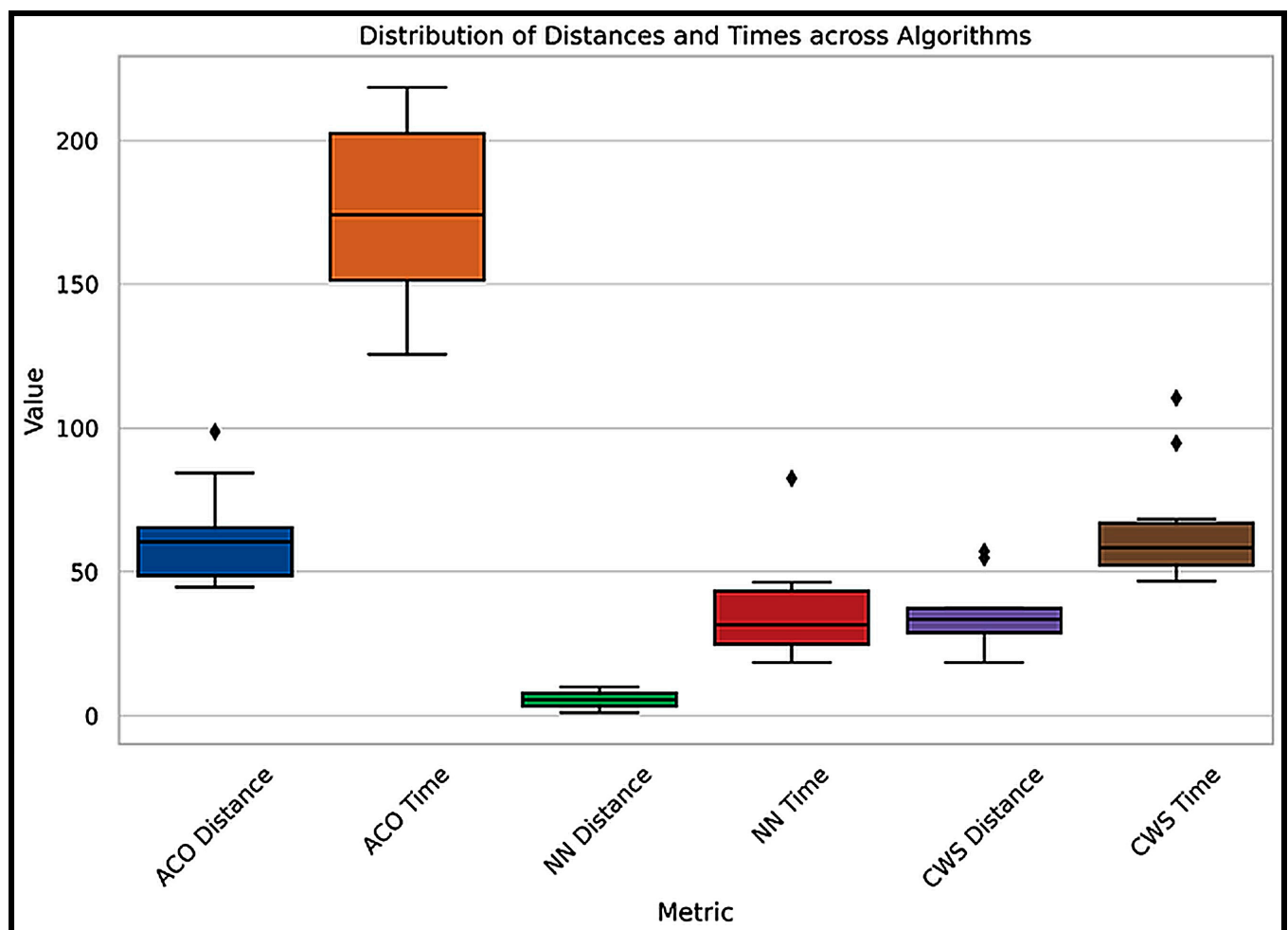


Figure 6. Distribution of distances and times across algorithms.

Figure 7 compares the efficiency of each algorithm across clusters using the distance-to-time ratio. Higher values indicate more distance covered per unit of time, representing better route compactness and temporal utilization. CWS consistently achieves the highest efficiency, especially in Clusters 5–7, reinforcing its balance between short routes and fast traversal. ACO shows moderate and stable efficiency but remains below CWS in all clusters. NN, while occasionally peaking in select clusters, displays lower and more erratic efficiency, reflecting its inconsistent performance. Overall, CWS delivers the most reliable and efficient routing outcomes.

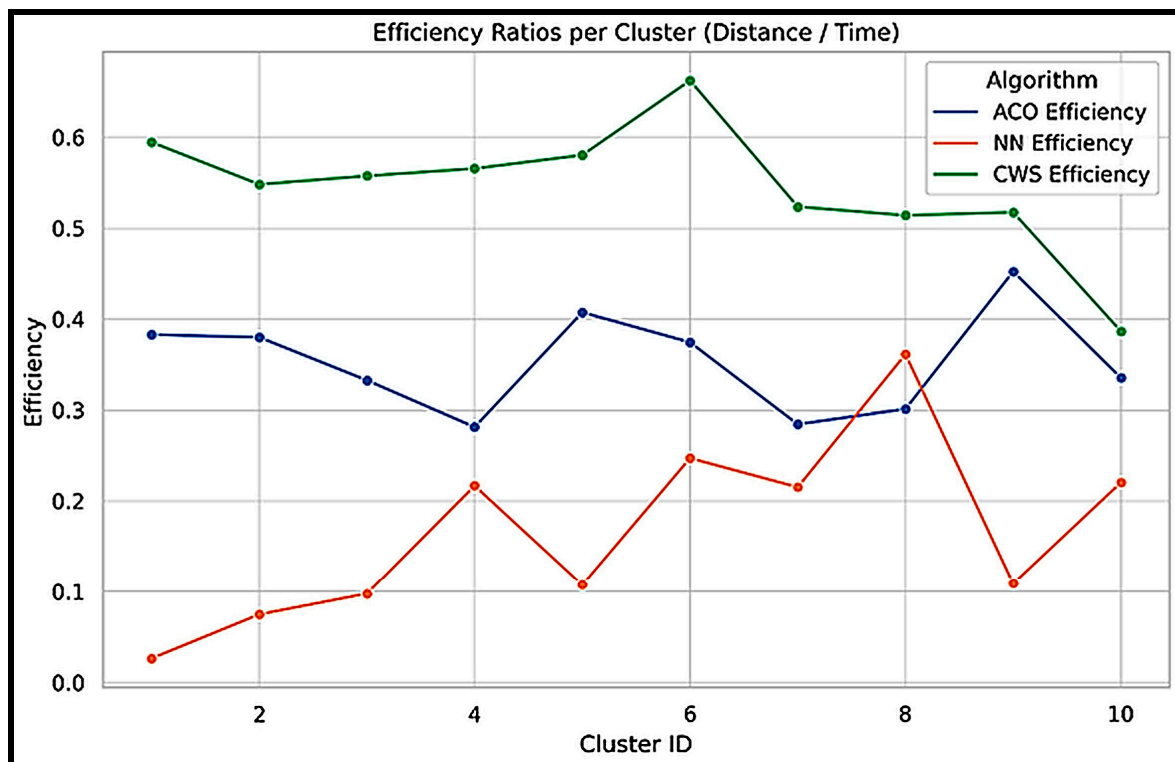


Figure 7. Efficiency ratios per cluster (distance/time).

To further evaluate algorithm performance under a multi-criteria framework, a performance quadrant plot was developed based on average distance and travel time in Figure 8. The plot is divided using the mean values of both metrics, forming four performance regions. Algorithms located in the lower-left quadrant (low distance and low time) represent superior operational efficiency. As shown in Figure 8, the Clarke–Wright Savings (CWS) algorithm falls within the optimal quadrant, indicating balanced minimization of both travel distance and time. The Nearest Neighbor (NN) algorithm demonstrates moderate performance, while Ant Colony Optimization (ACO) occupies the least efficient quadrant characterized by both high distance and high travel time. This visualization reinforces earlier findings that CWS provides the most reliable and efficient routing performance across clusters.

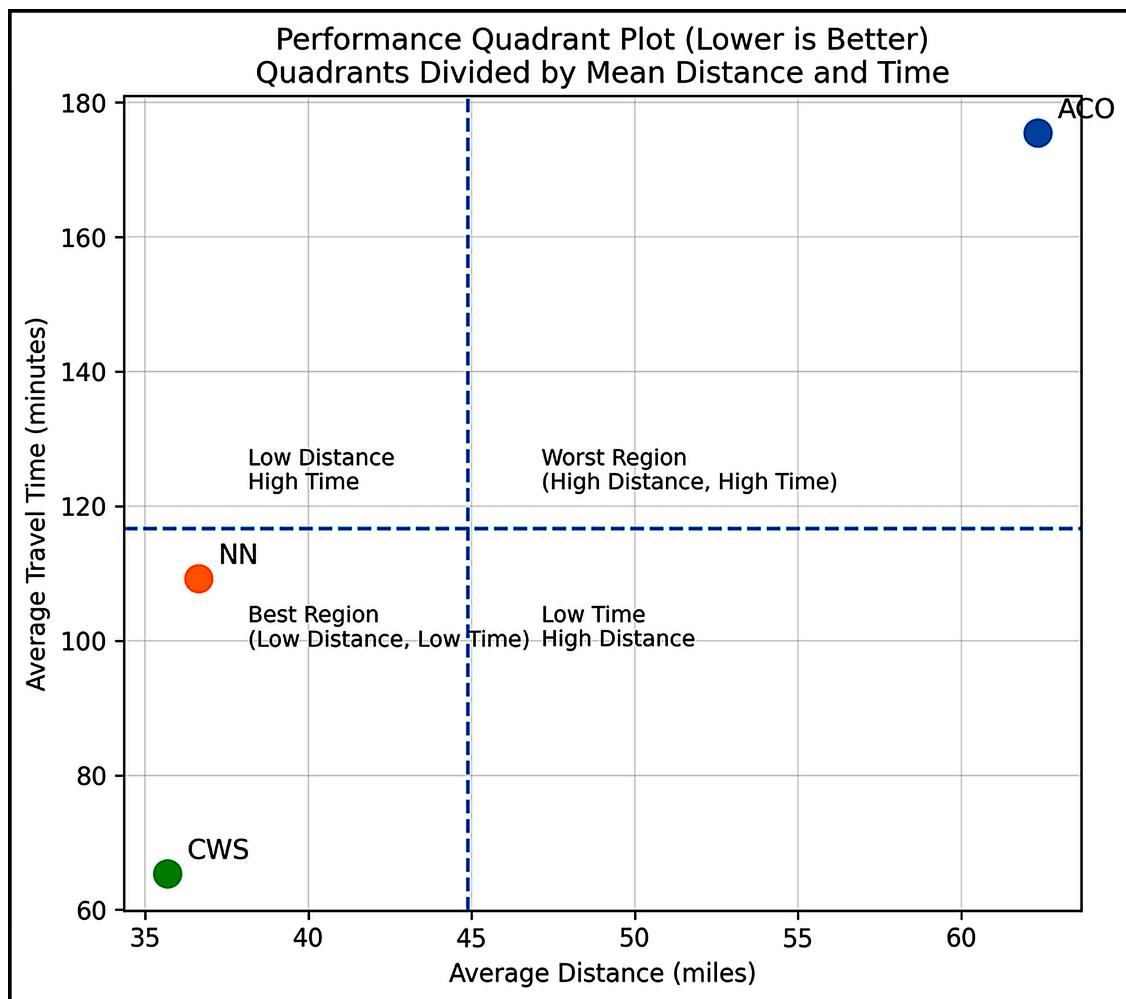


Figure 8. Performance quadrant plot (distance vs. time).

4.4. Composite Performance Index (CPI) Analysis

To jointly evaluate spatial and temporal routing efficiency, a Composite Performance Index (CPI) was defined by aggregating normalized travel distance and travel time. The CPI integrates both performance dimensions into a single scalar metric for comparative evaluation of routing algorithms.

For each cluster, normalization was performed independently across the three evaluated algorithms (NN, CWS, and ACO). For each metric x (distance or travel time), min-max normalization was applied as

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (13)$$

where $x_{\min}$ and $x_{\max}$ represent the minimum and maximum observed values of the metric among the three algorithms within that cluster. This transformation scales the values to the interval $[0, 1]$. Normalization was therefore cluster-specific, and CPI values represent relative performance within each cluster rather than absolute performance across clusters.

Because lower distance and travel time correspond to better routing performance, the normalized values were inverted prior to aggregation. The CPI was then computed as

$$\text{CPI} = \frac{(1 - D^*) + (1 - T^*)}{2} \quad (14)$$

where D^* and T^* denote the normalized distance and travel time, respectively.

The resulting CPI values for each algorithm are summarized in Table 4.

Table 4. Composite Performance Index (CPI) for routing algorithms.

Algorithm	CPI
ACO	0.00
NN	0.78
CWS	1.00

The CPI is introduced as a deterministic aggregation measure based on standard mathematical scaling and averaging operations. It is not proposed as a new theoretical optimization model, but rather as a transparent performance-comparison index constructed from normalized routing metrics.

Under this formulation:

- CPI = 1 indicates the best-performing algorithm within a cluster.
- CPI = 0 indicates the worst-performing algorithm within that cluster.
- Intermediate values reflect proportional relative performance.

The CPI results indicate that CWS achieved the highest composite performance among the evaluated algorithms, reflecting superior combined efficiency in travel distance and travel time under the defined modeling framework. NN demonstrated an intermediate performance, while ACO exhibited a comparatively lower composite efficiency across the evaluated clusters.

These observed rankings correspond to an incapacitated, distance-time minimization framework; incorporating additional operational constraints such as vehicle capacity limits, time windows, or service durations could influence route feasibility and potentially modify comparative outcomes.

Additionally, the relative performance of routing algorithms may vary under alternative cluster configurations, as changes in cluster size or spatial dispersion could influence route structure and algorithm behavior.

These findings are consistent with and extend the prior routing literature. The superior performance of the Clarke–Wright Savings (CWS) algorithm aligns with classical vehicle routing studies demonstrating the efficiency of savings-based constructive heuristics in depot-centered distribution systems [6,14,15]. The moderate but variable performance of the Nearest Neighbor (NN) heuristic corresponds with foundational analyses showing that greedy nearest-selection strategies often produce solutions significantly above optimal in larger TSP instances [5,11,12]. Although Ant Colony Optimization (ACO) has been shown to outperform traditional heuristics under dynamic or extensively tuned configurations [19–21], its comparatively lower standalone performance in this study suggests that under semi-static, traffic-informed cost matrices without adaptive parameter retuning, deterministic savings-based routing may provide more stable outcomes. Furthermore, unlike recent dynamic and heterogeneous VRP frameworks incorporating congestion feedback and operational constraints [22,23], the present study isolates foundational routing logic under identical real-world Amazon delivery data, thereby establishing a controlled empirical benchmark for heuristic performance in practical urban last-mile contexts.

5. Conclusions and Recommendations

This study evaluated three widely recognized routing optimization techniques, Nearest Neighbor (NN), Clarke–Wright Savings (CWS), and Ant Colony Optimization (ACO), using real-world Amazon last-mile delivery data from the Nashville metropolitan area. By framing the problem within the Traveling Salesman Problem (TSP) and analyzing ten distinct delivery clusters under traffic-informed conditions, this study provided a

comprehensive assessment of algorithmic efficiency, robustness, and operational practicality.

Across all clusters, the Clarke–Wright Savings Algorithm consistently demonstrated a superior performance in minimizing both total travel distance and travel time. In high-dispersion Cluster 9, CWS reduced total distance to 57.21 miles compared to 82.44 miles for NN and 98.77 miles for ACO, representing reductions of approximately 30% and 42%, respectively. Similarly, in Cluster 10, CWS achieved a total distance of 18.50 miles compared to 45.32 miles for NN, reflecting a reduction of nearly 59%. Travel time improvements were also substantial. In Cluster 1, CWS reduced total travel time to 61.48 min compared to 109.38 min for NN and 173.52 min for ACO, corresponding to reductions of approximately 44% and 65%, respectively. These consistent improvements across spatially diverse clusters demonstrate the algorithm's ability to compress both route length and duration under realistic roadway conditions.

The Nearest Neighbor algorithm exhibited a strong performance in compact clusters, such as Cluster 4 (18.43 miles and 58.48 min), but produced significantly longer routes in spatially dispersed clusters, most notably 82.44 miles and 196.48 min in Cluster 9. This variability underscores NN's sensitivity to spatial configuration and its susceptibility to local decision bias. Ant Colony Optimization demonstrated adaptive exploration behavior but produced the highest variability in both distance and time. In several clusters (e.g., 5, 7, and 9), ACO generated routes exceeding 200 min in duration and distances approaching 100 miles, indicating less stable convergence under the defined semi-static routing framework.

Beyond individual metric comparisons, the multi-criteria evaluation framework introduced in this study further strengthened these conclusions. Efficiency ratio analysis (distance-to-time) showed that CWS consistently achieved the highest compactness across clusters, particularly in Clusters 5–7. Distributional analysis using boxplots confirmed that CWS maintained the narrowest spread in both travel distance and travel time, demonstrating superior operational stability and predictability. Performance quadrant analysis placed CWS exclusively within the optimal low-distance, low-time region, while NN occupied an intermediate region and ACO consistently fell within the least efficient quadrant characterized by both high distance and high travel time.

The Composite Performance Index (CPI) analysis provided additional validation by consolidating normalized distance and travel time into a unified performance measure. CWS achieved the highest CPI value (1.00), compared to 0.78 for NN and 0.00 for ACO, confirming its balanced superiority across spatial and temporal dimensions. The alignment between direct route metrics, percentage improvements, efficiency ratios, distributional stability, quadrant positioning, and CPI ranking provides strong convergent evidence that CWS offers the most reliable routing solution under the evaluated urban delivery conditions.

While the findings strongly support the deployment of CWS in semi-static, depot-centered delivery systems, several limitations warrant consideration. The routing problems were formulated as incapacitated single-vehicle TSP instances; vehicle capacity limits, service times, delivery time windows, and multi-vehicle coordination constraints were not incorporated. Travel times were obtained using the Google Maps Distance Matrix API without departure time-specific congestion parameters, and therefore reflect typical rather than dynamic peak-period conditions. Delivery clusters were derived from operational Route IDs without independent clustering optimization. Additionally, this study did not benchmark against state-of-the-art exact solvers (e.g., Concorde, LKH) or advanced metaheuristics such as Genetic Algorithms or Variable Neighborhood Search, as the objective was controlled empirical benchmarking under identical traffic-informed cost inputs rather than global optimality validation.

To advance scalable and adaptive last-mile routing, future research should incorporate capacity and time window constraints, integrate departure time-dependent traffic modeling, evaluate alternative clustering strategies, and explore hybrid frameworks combining deterministic savings-based routing with adaptive metaheuristic refinement. Expanding benchmarking to include exact solvers and contemporary metaheuristics would further contextualize algorithmic performance relative to optimality bounds.

Overall, this research demonstrates that integrating real-world traffic-informed cost matrices with multi-criteria performance evaluation yields deeper insight into routing algorithm suitability for urban last-mile logistics. The combined use of direct route metrics, efficiency ratios, performance quadrant analysis, and composite indexing provides a robust and decision-oriented benchmarking framework. Under the evaluated conditions, the Clarke–Wright Savings Algorithm demonstrates superior stability, efficiency, and scalability, supporting its role as a practical baseline for urban last-mile delivery optimization.

Author Contributions: Conceptualization, A.Y., D.C. and M.R.; methodology, A.Y., D.C. and M.R.; validation, A.Y., D.C. and M.R.; formal analysis, A.Y., D.C. and M.R.; investigation, A.Y.; resources, D.C.; data curation, A.Y. and D.C.; writing—original draft preparation, A.Y. and D.C.; writing—review and editing, A.Y. and D.C.; visualization, M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Malshe Rohit was employed by the company Amazon Inc. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Wang, Y.; Zhang, D.; Liu, Q.; Shen, F.; Lee, L.H. Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions. *Transp. Res. Part E Logist. Transp. Rev.* **2016**, *93*, 279–293.
2. Gevaers, R.; Van de Voorde, E.; Vanelander, T. Cost Modelling and Simulation of Last-mile Characteristics in an Innovative B2C Supply Chain Environment with Implications on Urban Areas and Cities. *Procedia Soc. Behav. Sci.* **2014**, *125*, 398–411.
3. Applegate, D.; Bixby, R.; Chvátal, V.; Cook, W. *The Traveling Salesman Problem: A Computational Study*; Princeton University Press: Princeton, NJ, USA, 2011.
4. Dasgupta, S.; Papadimitriou, C.H.; Vazirani, U.V. Chapter 6 Dynamic Programming. Algorithms, 2006. Available online: <https://people.eecs.berkeley.edu/~vazirani/algorithms/chap6.pdf> (accessed on 25 January 2026)
5. Rosenkrantz, D.; Stearns, R.; Lewis, P., II. An Analysis of Several Heuristics for the Traveling Salesman Problem. *SIAM J. Comput.* **1977**, *6*, 563–581.
6. Clarke, G.; Wright, J.W. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Oper. Res.* **1964**, *12*, 568–581.
7. Dorigo, M.; Birattari, M.; Stutzle, T. Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. <https://doi.org/10.1109/CI-M.2006.248054>.
8. Nashville-Davidson—Murfreesboro—Franklin, TN Metro Area Census Data. Available online: <https://censusreporter.org/profiles/31000US34980-nashville-davidson-murfreesboro-franklin-tn-metro-area/> (accessed on 7 August 2025).
9. Available online: <https://www.census.gov/quickfacts/fact/table/nashvilledavidsonmetropolitangovernmentbalancetennessee/EDU635223?> (accessed on 7 August 2025).
10. Available online: <https://brightlio.com/nashville-data-centers/> (accessed on 7 August 2025).

11. Reinelt, G. *The Traveling Salesman: Computational Solutions for TSP Applications*; Springer: Berlin/Heidelberg, Germany, 1994.
12. Johnson, D.; McGeoch, L. *The Traveling Salesman Problem: A Case Study in Local Optimization*; John Wiley and Sons: London, UK, 1997; pp. 215–310.
13. Gendreau, M.; Potvin, J.-Y. Metaheuristics in Combinatorial Optimization. *Ann. Oper. Res.* **2005**, *140*, 189–213.
14. Gendreau, M.; Hertz, A. A Tabu Search Heuristic for the Vehicle Routing Problem. *Manag. Sci.* **1994**, *40*, 1276–1290.
15. Toth, P.; Vigo, D. *The Vehicle Routing Problem*; SIAM Monographs on Discrete Mathematics and Applications; Università Degli Studi di Bologna: Bologna, Italy, 2002.
16. Tan, K.; Lee, L.; Zhu, Q.; Ou, K. Heuristic methods for vehicle routing problem with time windows. *Artif. Intell. Eng.* **2001**, *15*, 281–295.
17. Osman, I.; Kelly, J. Meta-Heuristics Theory and Applications. *J. Oper. Res. Soc.* **1997**, *48*, 657.
18. Dorigo, M.; Maniezzo, V.; Colomi, A. Ant System: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B* **1996**, *26*, 29–41.
19. Dorigo, M.; Gambardella, L.M. Ant Colony System: A cooperative learning approach to the Traveling Salesman Problem. *IEEE Tr. Evol. Comp.* **1997**, *1*, 53–66.
20. Stützle, T.; Hoos, H.H. MAX-MIN Ant System. *Future Gener. Comput. Syst.* **2000**, *16*, 889–914.
21. Socha, K.; Dorigo, M. Ant colony optimization for continuous domains. *Eur. J. Oper. Res.* **2008**, *185*, 1155–1173.
22. Liu, L.; Wang, T. Optimizing urban last mile delivery efficiency through dynamic vehicle routing heuristics and traffic flow analysis. *Sci. Rep.* **2026**, *16*, 483.
23. Perić, N.; Begović, S.; Lešić, V. Optimization of Heterogeneous Last-Mile Delivery of Fresh Products Considering Traffic Congestions and Other Real-World Parameters. *IEEE Access* **2025**, *13*, 87193–87207. <https://doi.org/10.1109/ACCESS.2025.3569578>.
24. Shirmohammadi, H.; Najib, A.S.; Hadadi, F. Identification of Road Critical Segments Using Wavelet Theory and Multi-Criteria Decision-Making Method. *Eur. Transp.* **2018**, *68*.
25. Shirmohammadi, H.; Hadadi, F. APPLICATION OF FUZZY LOGIC FOR EVALUATION OF RESILIENT MODULUS PERFORMANCE OF STONE MASTIC ASPHALT. *J. Theor. Appl. Inf. Technol.* **2017**, *95*, 1–12.
26. Samadi, H.; Aghayan, I.; Shaaban, K.; Hadadi, F. Development of Performance Measurement Models for Two-Lane Roads under Vehicular Platooning Using Conjugate Bayesian Analysis. *Sustainability* **2023**, *15*, 4037.
27. Liu, F.; Lu, C.; Gui, L.; Zhang, Q.; Tong, X.; Yuan, M. Heuristics for Vehicle Routing Problem: A Survey and Recent Advances. *arXiv* **2023**, arXiv:2303.04147.
28. Parsusah, M.; Tlelo-Cuautle, E.; Lien, C.H.; Vaidyanathan, S.; Mamat, M.; Jerusalem, M.A.; Komaro, M. *The 3rd International Conference on Computer, Science, Engineering and Technology (ICComSET 2020)*; AIP Publishing: Melville, NY, USA, 2023.
29. Larson, R.C.; Odoni, A.R. MIT Urban Operations Research. Available online: https://web.mit.edu/urban_or_book/www/book/chapter6/6.4.12.html (accessed on 12 July 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.