Advances in Science and Technology
Research Journal

Advances in Science and Technology Research Journal, 2025, 19(11), 21–32
https://doi.org/10.12913/22998624/208843
ISSN 2299-8624, License CC-BY 4.0

Received: 2025.07.23
Accepted: 2025.09.15
Published: 2025.10.01

Challenges of delivery scheduling in real-world enterprise operations – a hybrid approach to transport management

Jerzy Merkisz[1], Anna Borucka[2], Patrycja Guzanek[3*]

[1] Faculty of Civil and Transport Engineering, Poznań University of Technology, ul. Piotrowo 3, Poznań, Poland
[2] Faculty of Security, Logistics and Management, Military University of Technology, ul. Gen. Sylwestra Kaliskiego 2, Warsaw, Poland
[3] Doctoral School of Military University of Technology, ul. Gen. Sylwestra Kaliskiego 2, Warsaw, Poland
* Corresponding author's e-mail: patrycja.guzanek@student.wat.edu.pl

ABSTRACT

Efficient delivery scheduling remains a key challenge in transport logistics, especially under real-world constraints such as vehicle capacity, time windows, traffic conditions, and weather. To address this, a hybrid metaheuristic algorithm combining Adaptive Large Neighborhood Search (ALNS) and tabu search was developed, where an initial greedy solution is iteratively improved through global diversification and local optimization. The approach balances solution quality and computational time by integrating broad search mechanisms with focused refinements. A case study using real company data validates the method's effectiveness in reducing route cost and improving operational efficiency. The results also highlight improved route structure and service consistency. This confirms the practical relevance of the model and its potential for broader application in logistics optimization.

Keywords: vehicle routing, hybrid metaheuristics, transport optimization.

INTRODUCTION

Efficient delivery scheduling in logistics and supply chain management remains a critical area of research, primarily due to its role in the "last mile" – the final and often most complex and expensive stage of product delivery to the end customer. Well-designed routing enables not only timely order fulfillment, which directly impacts customer satisfaction and loyalty, but also more effective use of company resources. In the context of growing e-commerce demand and increasing pressure to shorten delivery times, the ability to dynamically and optimally plan routes has become a key factor in achieving competitive advantage. Route optimization also influences strategic decisions, including fleet sizing and distribution center location. Consequently, the scheduling process has a significant impact on operational costs, with transportation being one of the largest expense categories in logistics budgets, driven by fuel, driver wages, and vehicle maintenance. Reducing total route distance leads directly to lower fuel consumption and decreased vehicle wear, while minimizing total driver working time reduces labor costs, including overtime. Moreover, efficient planning can reveal that fewer vehicles are needed to serve the same number of customers, resulting in capital and maintenance savings. Even the implementation of simple optimization algorithms can lead to substantial cost reductions. From an operational efficiency perspective, optimized scheduling maximizes resource utilization by improving vehicle load rates, reducing empty mileage, and increasing route profitability. It also enables service of more customers in less time or within narrower delivery time windows, increasingly demanded by the market. Improved efficiency also translates into environmental benefits – shorter routes result in lower emissions, aligning with sustainability goals and strengthening corporate image. As a result, delivery scheduling is no longer viewed solely as a technical optimization challenge, but as a strategic tool for building efficient, cost-effective, and socially responsible logistics systems.

In response to these challenges, the aim of this study was to design, implement, and evaluate the effectiveness of a hybrid metaheuristic for solving the vehicle routing problem. Specifically, the goal was to investigate how the combination of two distinct optimization strategies – global exploration of the solution space and local, fine-tuned improvements – affects the quality and speed of route generation. The research hypothesis assumed that integrating large, diversified changes (the destroy and repair operators in the Adaptive Large Neighborhood Search – ALNS algorithm) with a local intensification mechanism (tabu search) would yield better results than applying either strategy in isolation. The proposed hybrid algorithm is expected to deliver higher-quality solutions (i.e., shorter routes) within competitive computational times, outperforming standard metaheuristics such as the classic genetic algorithm, and offering more practical applicability than exact methods (solvers) for realistically scaled problems.

This study directly addresses a persistent and significant research gap in the field of logistics optimization. Although the vehicle routing problem (VRP) and its variants have been extensively studied for decades, and metaheuristics such as genetic algorithms and tabu search are well documented, the gap lies not in the problem itself, but in the methods used to solve it. Therefore, the scientific contribution of this research can be defined as follows:

- A response to the need for developing integrated solution approaches and evaluating which combinations of mechanisms yield the best results. The proposed method in this article tests a specific combination: the strength of ALNS, aimed at escaping local optima through substantial route restructuring, and the precision of tabu search in refining local segments of the solution.
- Consideration of a practically relevant trade-off in logistics between finding an ideal route and the computational time available to search for it – aiming for the best balance between solution quality and computational effort.

The main scientific contribution of this study can be identified as follows:

- A contribution to the advancement of knowledge on designing effective metaheuristics through the proposal and evaluation of a specific algorithmic architecture – not a simple combination of two methods, but a well-structured integration in which one mechanism (large-scale changes) creates space for the other (local refinements) to operate effectively.
- Providing evidence of the effectiveness of the proposed algorithms based on real-world data from a distribution company, thus creating a valuable benchmark for potential users - both researchers and logistics enterprises with similar operational profiles.
- The results of the study also have a practical dimension, demonstrating that investment in the development of a more complex hybrid algorithm can yield tangible benefits in the form of reduced transportation costs (shorter routes) and improved operational efficiency (faster planning).

BACKGROUND

The delivery scheduling problem, widely recognized in academic literature as the VRP, is one of the most fundamental and frequently analyzed topics in logistics and operations research [1]. In its classical form, it involves determining an optimal set of routes for a fleet of vehicles that, starting from one or more depots, must serve a group of geographically dispersed customers and then return to the starting point. The objective is to minimize the total operational cost, typically associated with the overall distance, travel time, or number of vehicles used [2]. This problem is a generalization of the well-known Traveling Salesman Problem (TSP), with the key distinction that VRP involves multiple vehicles with limited capacity [3]. Due to its computational complexity – it belongs to the class of NP-hard problems – finding an optimal solution for real-life, large-scale instances is extremely challenging [4]. In general, VRP solution methods are classified into three main categories: exact algorithms, heuristics, and metaheuristics. Each of these approaches has different characteristics, making them suitable for different problem scales and precision requirements [5]. Selecting the appropriate method is a key trade-off between solution quality and the time required to find it [6, 7].

Exact algorithms, such as integer programming, branch-and-bound, and column generation, offer one fundamental advantage – they guarantee optimal solutions [8, 9]. These methods ensure that the obtained routing plan is the best possible with respect to a defined objective (e.g., minimum distance). They also serve as valuable benchmarks for evaluating the quality of faster, approximate methods. However, their main drawback – and a major limitation in practical applications – is computational complexity [10]. The time required to find a solution increases exponentially with problem size. In practice, this means that exact algorithms are capable of efficiently solving only small problem instances, typically involving a few dozen delivery points. For large logistics networks with hundreds or thousands of customers, their application becomes infeasible due to unacceptable computation times [11].

An alternative approach is heuristics – simple, intuitive construction methods that allow for very fast generation of good, though not necessarily optimal, solutions. Classic examples include the Clarke and Wright savings algorithm [12], the sweep algorithm [13], and the nearest neighbor method [14]. Their greatest advantages are speed and ease of implementation. They can quickly generate reasonable routing plans even for very large problems. However, they do not guarantee solution quality. Because heuristics make locally optimal choices at each step, they often become trapped in local optima, and the final result may be far from the best possible. Nevertheless, due to their speed, they are frequently used in systems that require real-time or dynamic planning, or as a starting point for more advanced optimization techniques.

Metaheuristics attempt to combine the strengths of both approaches, offering a compromise between solution quality and computational effort. These strategies guide simpler heuristics to more effectively explore the solution space and avoid local optima traps. Among the most popular metaheuristics used for VRP are those applied in this paper – tabu search – as well as simulated annealing, genetic algorithms [15, 16, 17], and ant colony optimization [18]. Their advantage lies in the ability to find very high-quality solutions, often optimal or near-optimal, for large and complex problems with multiple constraints. This comes at the cost of greater implementation complexity and the need to tune numerous control parameters, which can be time-consuming. Nevertheless, metaheuristics have become the most commonly used approach for solving real-world VRP instances in industry.

MATERIALS AND METHODS

As highlighted in the introduction, distribution companies are faced daily with complex decisions regarding which vehicle should be dispatched, in what order deliveries should be made, and along which route – all while ensuring timely delivery and minimizing travel distance. An optimal routing solution helps reduce fuel consumption, limits vehicle wear, and enhances customer service. In this study, a routing model was proposed for a transport company and its fleet, with the goal of delivering packages within specified time windows while minimizing distance and travel time. The dataset used in the study was based on real operational data obtained from a logistics company, including exact delivery times and locations (GPS coordinates), parcel weight and volume, time windows for delivery, service duration, as well as contextual information such as traffic congestion based on time of day and weather conditions.

To solve the problem, a hybrid algorithm was implemented. First, a fast initial solution is generated, which is then improved through destroy and repair operations within the Adaptive Large Neighborhood Search (ALNS) framework. Finally, local route refinements are applied using the tabu search algorithm. The underlying assumption is that combining large-scale solution modifications with fine-grained local improvements helps avoid suboptimal local minima and leads to more effective routing outcomes.

Delivery scheduling is formulated as a vehicle routing problem with time windows (VRPTW). Let D = {1,2, ..., n} denote the set of delivery points, and V = {1,2, ..., m} the set of vehicles. Each delivery point i ∈ D is associated with: a weight w_i a volume v_i, and a delivery time window $[t_i^{min}, t_i^{max}]$. Each vehicle k ∈ V has a load capacity of W_k (kg) and a volume capacity of V_k (m^3). The objective is to minimize the total distance traveled by the fleet while applying penalties for time window violations.

The definition of the decision variable:

$$x_{i,j}^{k} = \begin{cases} 1, \text{if vehicle } k \text{ travels from from } i \text{ to } j \\ 0, \text{otherwise} \end{cases} \quad (1)$$

where: $i, j \in D \cup \{0\}$ and „0” denotes the depot.

Let d_{ij} be the distance between points t_{ij} – the travel time from i to j. If vehicle k visits point i at time $T_i^k > \delta_i^k$, a penalty P is applied, where:

$$\delta_i^k = \max\{0, T_i^k - \mathrm{t}_\mathrm{i}^{\max}\}.$$

The objective is to minimize the total travel distance plus the penalties for time window violations:

$$\min \sum_{k\in V} \sum_{i\in D\cup\{0\}} \sum_{j\in D\cup\{0\}} d_{ij}\, x_{i,j}^k + + P \sum_{k\in V} \sum_{i\in D} \delta_i^k \quad (2)$$

Subject to the following constraints:

- Each delivery point is visited exactly once:

$$\sum_{k\in V} \sum_{j\in D\cup\{0\}} x_{i,j}^k = 1 \; \forall\; i \in D \quad (3)$$

- Depot flow balance is ensured:

$$\sum_{j\in D} x_{0,j}^k = \sum_{i\in D} x_{i,0}^k \, \forall\, k \in V \quad (4)$$

- Route consistency is ensured (flow conservation at each node):

$$\sum_{i\in D\cup\{0\}} x_{i,h}^k = \sum_{j\in D\cup\{0\}} x_{h,j}^k \,\forall\, h \in D \cup \{0\}, \forall\, k \in V \quad (5)$$

- Load capacity constraint:

$$\sum_{i\in D} w_i \sum_{j\in D\cup\{0\}} x_{i,j}^k \le W_k \quad (6)$$

$$\sum_{i\in D} v_i \sum_{j\in D\cup\{0\}} x_{i,j}^k \le V_k \,\forall\, k \in V \quad (7)$$

- Arrival time update (big-M method)

$$\text{If } x_{i,j}^k = 1 \text{ then } T_j^k \ge T_i^k + t_{ij} - - M\left(1 - x_{i,j}^k\right) \forall\, i,j \in D \cup \{0\}, \forall\, k \in V \quad (8)$$

where: M is a sufficiently large constant – a numerical value large enough not to restrict the model unintentionally, yet not excessively large to avoid numerical instability.

- Time Windows:

$$T_i^k \ge t_i^{min} \;\; , T_i^k \le t_i^{max} + \delta_i^k, \; \delta_i^k \ge 0 \,\forall\, i \in D, \forall\, k \in V \quad (9)$$

In the first stage of generating the initial solution S_0 a greedy algorithm was applied. Initially, the set of all unserved delivery points is denoted as $U \leftarrow D$. Then, for each vehicle $k \in V$, we initialize its route starting from the depot. The remaining available weight and volume capacities are set to the vehicle's full capacity: $W_k^{poz} \leftarrow W_k$, $V_k^{poz} \leftarrow V_k$. In the greedy loop, we iteratively select the point $i^* \in U$ that minimizes the distance from the current position of the vehicle:

$$i^* = \arg\min_{i\in U} d_{c,i} \quad (10)$$

where: c represents the current position of the vehicle (with c = 0 referring to the depot).

After adding point i^* to the route of vehicle k, update the current position: $c \leftarrow i^*$.

Before assigning point i^* check whether the expected arrival time T_i^k fits within the time window $[\mathrm{t}_\mathrm{i}^{\min}, \mathrm{t}_\mathrm{i}^{\max}]$. If not, the point is skipped and the next closest one is considered. Once a feasible i^* is found, the algorithm checks whether its demand w_i and v_i do not exceed the remaining capacities W_k^{poz} and V_k^{poz}. If the condition is met, the point is assigned to vehicle k, and capacities are updated:

$$W_k^{poz} \leftarrow W_k^{poz} - w_i, \quad V_k^{poz} \leftarrow V_k^{poz} - v_i \quad (11)$$

Then, point i^* is removed from the set U. Otherwise, the construction of the current route for vehicle k is terminated, and the algorithm proceeds to the next vehicle. This procedure is repeated until the set U is empty or none of the vehicles have sufficient resources to serve the remaining requests.

In the next stage of the study, the solution S is improved using an iterative ALNS framework. In ALNS, the weights of the operators are updated in each iteration based on their performance, increasing the likelihood of selecting those that have produced better results in previous iterations. Each iteration involves a Destroy and Repair phase. In the Destroy phase, a destroy operator d is selected from the pool D and a subset of delivery points is removed from the current solution:

$$R = d(S), S' = S \setminus R. \quad (12)$$

In the destroy phase, three basic removal operators were applied, each differing in the mechanism used to select elements for reinsertion – namely: remove-random, remove-worst, and remove-TW-violations. The remove-random operator randomly removes a subset $R_{rand} \subset S$ of a size *k*, where *S* denotes the set of all requests in the current solution:

$$R_{rand} \sim Uniform\{M \subset S : |M| = k\} \quad (13)$$

This helps to avoid getting trapped in local minima. The remove-worst operator focuses on the requests that contribute the most to the total route cost. For each $i \in S$ the cost increase caused by serving that point is calculated as follows:

$$\Delta c_i = c(S) - c(S \setminus \{i\}) \quad (14)$$

where: c(·) denotes the total cost of the routes (e.g., total distance or travel time).

Next, we select:

$$R_{worst} = \underset{R \subset S, |R|=k}{arg\,max} \sum_{i \in R} \Delta c_i \ , \quad S \leftarrow S \setminus R_{worst} \quad (15)$$

which results in the removal of those *k* requests whose presence has the most negative impact on the solution's efficiency. The remove-TW-violations operator identifies requests that violate time windows and removes them, allowing for their reinsertion later in a schedule – compliant manner. For each $i \in S$ a violation index is defined as:

$$\chi_i = \begin{cases} 1, & t_i < t_i^{min} \ \lor \ t_i > t_i^{max} \\ & 0, otherwise \end{cases} \quad (16)$$

where: $[t_i^{min}, t_i^{max}]$ denotes the allowed time window, and t_i – is the expected service time.

The set of all points with $X_i = 1$ is denoted as V_{TW}. If $|V_{TW}| \leq k$, all elements of this set are removed; otherwise, a subset V_{TW} choose the subset R_{TW} of size *k* is selected and the following operation is performed:

$$S \leftarrow S \setminus R_{TW} \quad (17)$$

Each of these operators determines the degree of 'destruction' in a different way, setting the stage for the repair phase, in which the removed requests are reinserted using dedicated repair operators. This mechanism enables effective exploration of the solution space and contributes to reducing the total cost.

The repair phase involves selecting a repair operator and reinserting the removed points *R* back into the routes:

$$S'' = r(S', R) \quad (18)$$

In the repair phase of the ALNS algorithm, the points removed during the destroy step are reinserted into the solution. The selection among available repair operators is made adaptively, with operator usage frequency adjusted based on their past effectiveness. This study implemented three key repair operators, described below.

The greedy-insert operator assumes that each removed point is reinserted into the position that causes the smallest possible increase in total cost (i.e., distance). Let the current route of vehicle *k*, including the depot at the start and end, be represented as:

$$\pi^k = \left(\pi_0^k, \pi_1^k \dots, \pi_{n_k}^k, \pi_{n_{k+1}}^k\right) \quad (19)$$

where: $\pi_0^k, \dots, \pi_{n_{k+1}}^k = 0$ and „0" denotes the depot index.

Let *R* be the set of removed points. For each $i \in R$, vehicle rout *k* and possible insertion position $p \in \{0, 1, \dots, n_k\}$ define:

$$a = \pi_p^k \quad (20)$$

$$c = \pi_{p+1}^k \quad (21)$$

for each insertion position, the cost increase is calculated as:

$$\Delta_{i,k,p} = \left(d_{a,i} + d_{i,c}\right) - d_{a,c} \quad (22)$$

where: *a* is the predecessor and *c* is the successor on the route, we select route k^* and position p^*, for which $\Delta_{i,k,p}$ is minimal, and insert point *i* into route k^* at position p^*.

This procedure is repeated independently for all $i \in R$, ensuring that each point is inserted where it causes the smallest cost (distance) increase.

The regret *k* operator accounts for the fact that some points may have multiple good insertion options. It prioritizes those for which the "loss" due to delayed insertion is greatest. For each removed point $i \in R$ and each route *k* compute the vector of all insertion costs: $\{c_{i,k,p}\}_{p=0}^{n_k}$, where $c_{i,k,p} = (d_{a,i} + d_{i,c}) - d_{a,c}$. All possible insertions are then sorted in ascending order: $c_{i,k}^{(1)} \leq c_{i,k}^{(2)} \leq \cdots c \leq_{(i,k)}^{(j)}$. For each point *i* select the *k* smallest values and define the $regret_i$:

$$regret_i = \sum_{j=2}^{k} \left(c_{(i,*)}^{(j)} - c_{(i,*)}^{(1)} \right) \quad (23)$$

where: $c_{i,*}^{(j)}$ – the j-th smallest insertion cost among all possible routes.

Among all points, select the one with the highest $iregret_i$ and insert it at the position corresponding to $c_{i,*}^{(1)}$. Point i is then removed from set R and the process is repeated until the set is exhausted. As a result, points with fewer good insertion options (high $regret_i$ are prioritized, preventing the 'blocking' of suitable positions later in the procedure. The fastest-insert operator is a variant of the greedy-insert, but instead of minimizing the increase in distance, it focuses on minimizing the increase in travel time. This is done using the travel time matrix $T = [t_{x,y}]$, which was generated based on historical traffic and weather data:

$$t_{x,y} = time_{mat_{s[x,y]}} \quad (24)$$

For each $i \in R$, route k and position $p \in \{0, ..., n_k\} p \in \{0, \dots, n_k\} p \in \{0, \dots, n_k\}$ calculate:

$$\Delta_{i,k,p}^{time} = (t_{a,i} + t_{i,c}) - t_{a,c} \quad (25)$$

Select the pair (k^*, p^*), that minimizes $\Delta_{i,k,p}^{time}$.

$$(k^*, p^*) = \arg\min_{k,p} \Delta_{i,k,p}^{time} \quad (26)$$

Next, point i is inserted at position p^* in route k^*. The entire procedure is repeated for the remaining points in R. As a result, the Fastest-Insert operator places each point where it causes the smallest increase in total travel time, thereby optimizing the routes in terms of cumulative travel time rather than just total distance.

To avoid getting trapped in local minima, a tabu search algorithm is applied after each Repair phase. Let S'' be the solution obtained after reinserting all points during the repair phase. Its neighborhood is generated using the swap operation – which involves exchanging two arbitrary requests i,j belonging to different routes (possibly also assigned to different vehicles). Each swap move (i,j) is recorded in the tabu list as a pair (i,j) with a fixed length L_{max}, meaning that reversing this move is prohibited for the next L_{max} iterations.

Among all tabu-restricted moves that improve the solution cost, the move that minimizes the objective function is selected and executed:

$$\Delta c_{i,j} = Cost(S''_{swap(i,j)}) - Cost(S'') \quad (27)$$

In this way, a new solution S''. is obtained. If its cost is lower than the current best solution S_{best}, an update is performed: $S_{best} \leftarrow S''$. Each S'' is locally improved, and the tabu list enforces exploration of new regions in the solution space.

The final step is weight adaptation. An adaptive frequency adjustment is applied to the use of operators from sets D (destroy) and R (repair). Each operator $o \in D \cup R$ is assigned a weight w_o, and its probability of being selected in a given iteration is proportional to w_o. After completing a full iteration – that is, obtaining a new solution S_{new} (after the tabu search phase) the cost change is calculated relative to the best solution found so far:

$$\Delta = Cost(S_{\text{new}}) - Cost(S_{best}) \quad (28)$$

Operators that participated in the current iteration are awarded a score based on Δ, according to the following rule:

$$ro = \begin{cases} \sigma_1, when\ \Delta < 0 - \ improvement \\ \sigma_2, when\ \Delta = 0 - \ definitive\ draw \\ \sigma_3,\ when\ \Delta > 0 - \ deterioration \end{cases} \quad (29)$$

where: $\sigma_1 > \sigma_2 > \sigma_3 \geq 0$ are predefined constants. The operator's weight is updated according to the following scheme:

$$w_o \leftarrow \rho\, w_o + (1 - \rho)\, r_o \quad (30)$$

where: $p \in [0,1]$ is the forgetting factor. After each adaptation, the selection probabilities of the operators are normalized:

$$p_o = \frac{w_o}{\sum_{o' \in D \cup R} w_{o'}} \quad (31)$$

Thanks to this mechanism, the algorithm increasingly favors those operators in subsequent iterations that contribute to actual solution improvement, while still maintaining a degree of exploration.

This combination of large-scale route modifications (ALNS) with local refinements (tabu search) enables the algorithm to escape local minima and achieve shorter, more efficient routes while satisfying vehicle capacity and time window constraints. Additionally, the adaptive weighting of operators allows the algorithm to "learn" which moves are most promising, accelerating convergence.

DELIVERY SCHEDULING – CASE STUDY

For the purpose of this study, the parameters of the ALNS algorithm with tabu search were defined. It was assumed that ALNS would perform up to 1000 iterations in search of improved solutions, while the local tabu search procedure would run 2000 iterations for each local improvement attempt. A given move (e.g., reassigning a task) would be considered 'tabu' for 400 iterations to avoid cycling, and in each tabu search iteration, 30 potential candidate moves would be evaluated.

In the first step, an initial greedy solution was developed according to the scheme shown in Figure 1. This solution assigns tasks to vehicle routes based on simple heuristic rules – such as always selecting the nearest available delivery point. This approach enables rapid generation of a feasible, though not necessarily optimal, solution, which serves as a starting point for further optimization. The results obtained also serve as a benchmark for comparison with the improved algorithm.

For the greedy algorithm, the total distance traveled by the fleet amounted to 3 063 917 km, providing a baseline for improvement using more advanced optimization methods such as ALNS and tabu search.

Next, route optimization was performed according to the scheme in Figure 2, using the ALNS algorithm, which dynamically selects destroy and repair operators. In each iteration, the algorithm removed selected orders from routes (e.g., randomly or based on cost) and attempted to reinsert them in a more optimal manner. This process was supported by a local improvement procedure, helping the algorithm avoid local optima. As a result, a significantly better solution was achieved – the total fleet distance was reduced to 1 167 352 km, representing a substantial improvement over the initial solution.

Figure 3 presents a convergence plot showing how the solution cost evolved across iterations of the ALNS + tabu search algorithm. Downward trends indicate solution improvements (lower cost), while flat segments correspond to iterations without improvement. A sharp drop in cost is visible during the initial phase (~first 100 iterations), where the algorithm quickly identifies much better solutions than the initial greedy one. During the mid-phase (~iterations 100–400), improvements continue but at a slower, less regular pace with occasional fluctuations. In the final phase

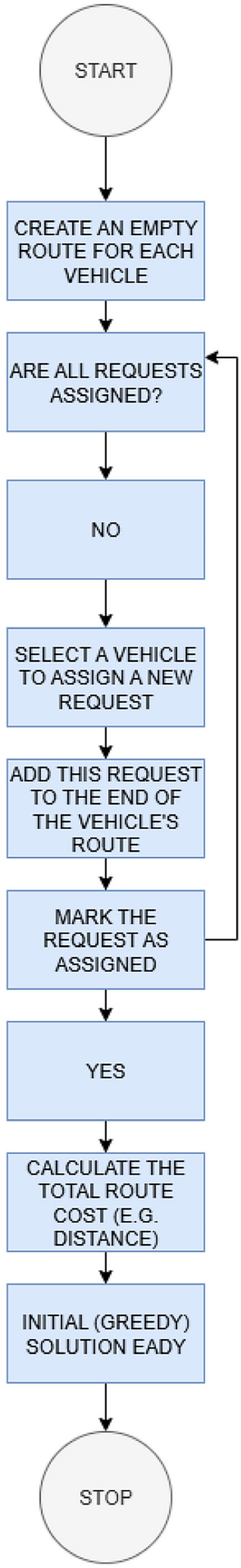

Figure 1. Greedy algorithm workflow

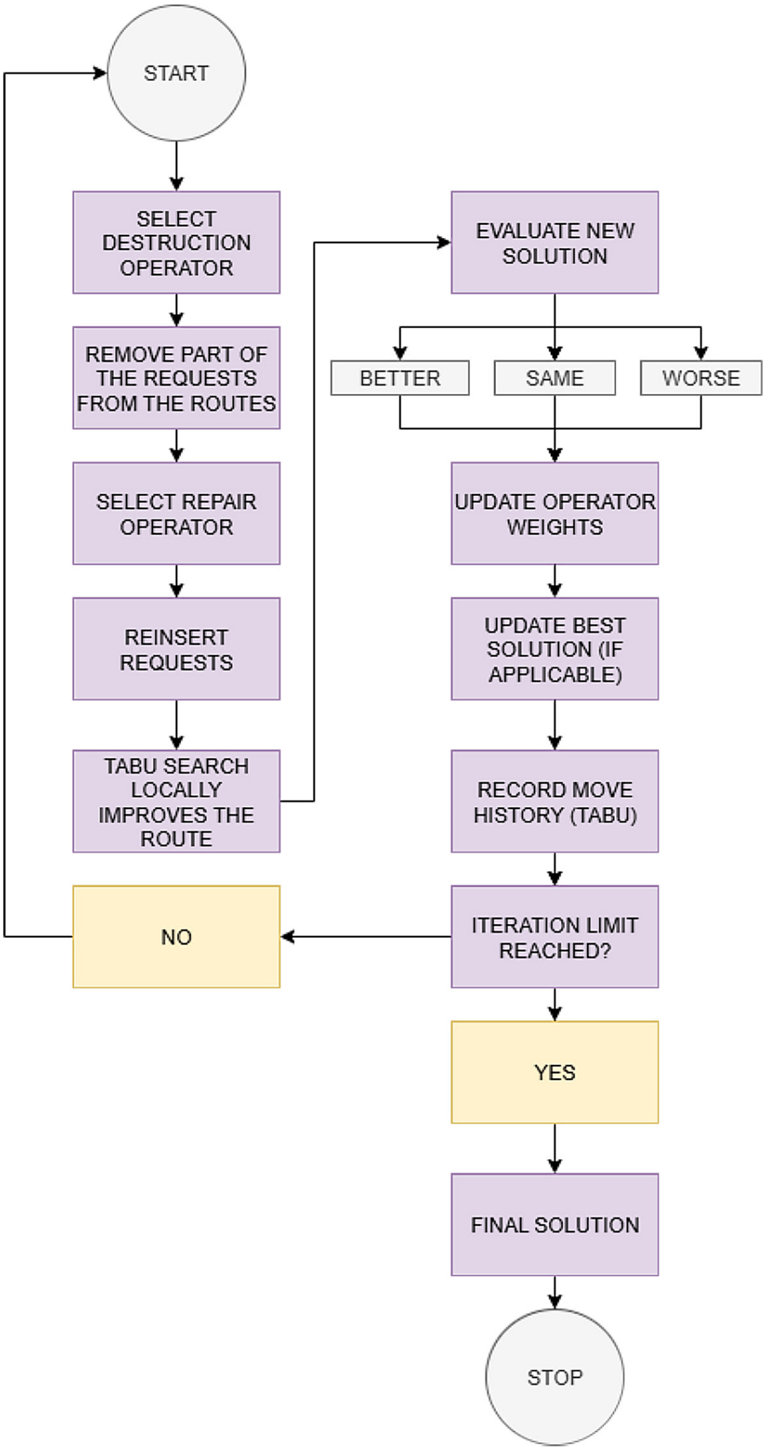

Figure 2. ALNS + tabu search workflow

(~iterations 400–1000), the cost stabilizes near a local optimum, indicating that further major improvements become harder to obtain. Figure 4 illustrates the behavior of the individual algorithms across iterations.

It can be observed that the blue line representing tabu search lies below the red ALNS line in most iterations. This suggests that local optimization consistently improves solution quality, smoothing out extreme cost fluctuations and leading to a more stable result. Despite some volatility, a clear downward trend in cost and convergence toward an optimal solution is evident.

Figure 5 compares vehicle routes at two stages of the algorithm. The right-hand side shows the initial solution generated by the greedy algorithm, while the left-hand side presents the final optimized solution.

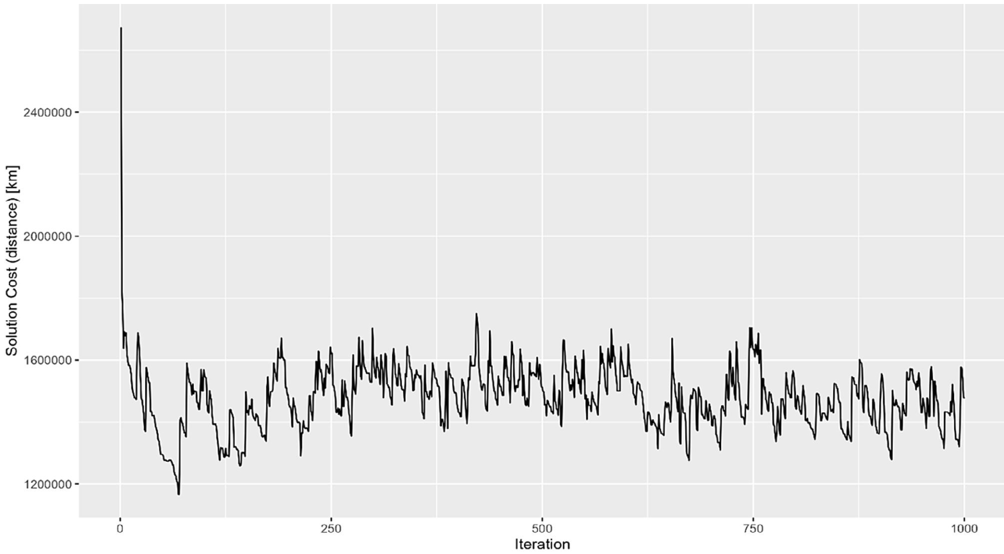

Figure 3. ALNS + tabu search cost trajectory

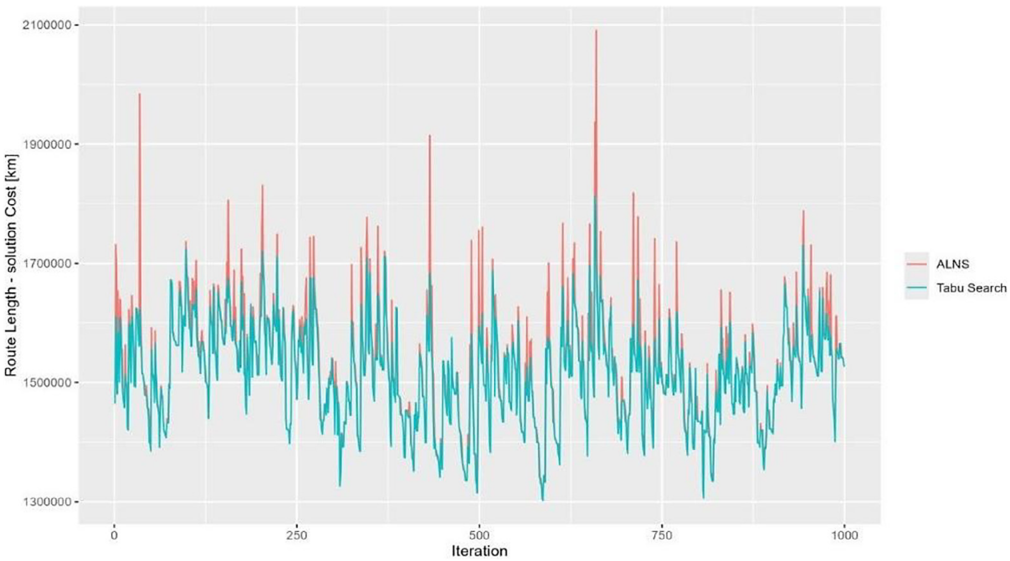

Figure 4. Cost trajectory: ALNS vs tabu search

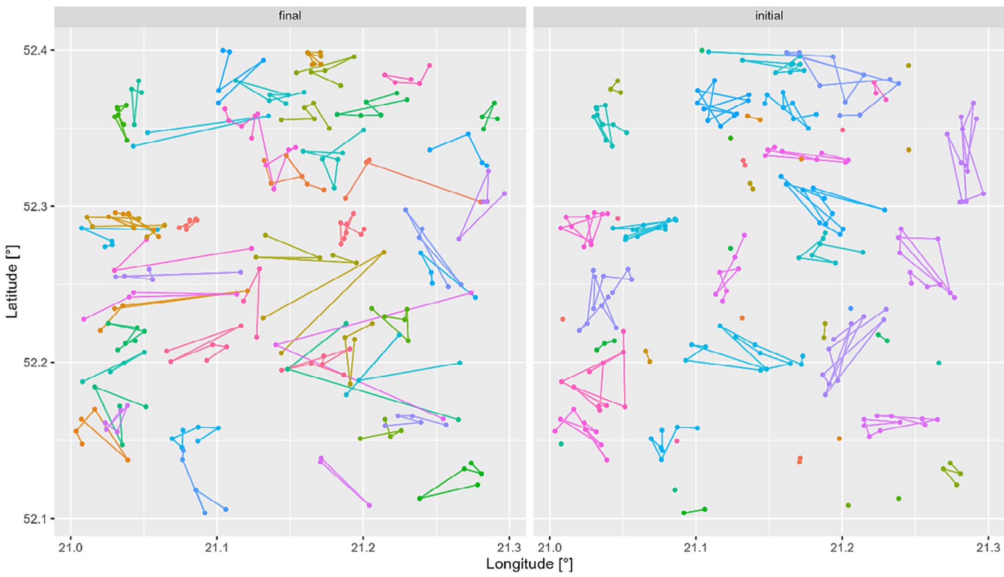

Figure 5. Route comparison: a) left-hand side: final optimized solution, b) right-hand side: initial greedy routes

The chart compares vehicle routes before ("initial") and after ("final") optimization using the ALNS algorithm combined with tabu search.

In the initial solution, routes are disorganized and scattered – many intersect, and vehicles serve geographically distant delivery points. Several one-stop routes are visible, indicating inefficient task assignment and limited route consolidation.

In contrast, the final (optimized) solution features more compact, spatially coherent routes. Vehicles serve clusters of nearby points, leading to shorter total distances and reduced overlap between routes. Notably, no one-stop routes are present in the optimized solution, reflecting better use of available resources and improved planning.

The number of serviced delivery points is identical in both solutions (242), confirming that all orders were fulfilled. However, the significant improvement in route structure demonstrates the effectiveness of the optimization process.

In conclusion, combining ALNS with tabu search significantly enhances solution quality. Tabu search compensates for the limitations of the greedy approach, yielding shorter, more organized, and operationally efficient routes. This validates the benefit of hybrid optimization strategies for complex routing problems.

DISCUSSION

The results of the conducted study clearly indicate that the applied route planning approach in the logistics company has yielded significant benefits. Most notably, the total distance traveled by the fleet was substantially reduced from over three million kilometers to just over one million. This reduction translates into measurable savings in fuel costs and driver working hours, as well as potentially reduced vehicle wear and lower environmental impact of transport operations.

The operation of the implemented solution also deserves attention. Even at an early stage of planning, it was possible to rapidly identify significantly better routes than those generated using simple heuristic rules. Subsequent improvements occurred gradually, primarily through minor route refinements that led to better organization and reduced travel time. The final outcome was visible not only in the numerical indicators but also in the structural characteristics of the routes – these became more coherent, localized, and logically organized, avoiding unnecessary intersections and redundant travel segments.

The study also confirmed that combining different types of route optimization techniques – both major restructurings and minor adjustments – delivers better results than relying on a single approach. On one hand, it enabled the reconfiguration of inefficient segments, while on the other, it allowed for the fine-tuning of segments that were already functioning effectively. This dual-level optimization approach helped prevent the routes from settling into suboptimal configurations with limited potential for further improvement [19].

Importantly, despite the substantial improvement in route quality, all delivery orders continued to be fulfilled – meaning that efficiency gains were not achieved at the expense of service coverage.

It is recommended to further develop the model by incorporating additional optimization criteria – such as driver labor costs, customer delivery time preferences, emissions levels [20], and loading and unloading times. A multi-criteria optimization framework could better reflect real business conditions and enhance the acceptability of the generated routes among logistics system users.

Future implementations should also consider integrating the algorithm with transportation management systems (TMS) [21], enabling dynamic route updates in response to real-time changes in road conditions or resource availability. Furthermore, adopting an online or hybrid distributed version of the algorithm could accelerate its performance and allow for real-time application [22].

Furthermore, optimized delivery scheduling contributes not only to reduced emissions and fuel use but also to lower urban noise pollution, particularly in densely populated areas [23, 24]. This is increasingly relevant as sustainable urban logistics systems seek to minimize both environmental and social externalities. Moreover, micro-mobility solutions - such as electric cargo bikes or small autonomous delivery vehicles - offer a promising complement to hybrid routing strategies, especially in addressing last-mile delivery challenges. Their integration into optimization frameworks could significantly enhance the flexibility and efficiency of delivery systems in urban settings [25].

It should be noted that the analysis was conducted based on data from a single transport company. Although the data were real and highly detailed, this limitation may affect the generalizability of the results to other industries, regions, or logistical models. Therefore, it would be beneficial to extend the study to companies with a broader scope of operations.

CONCLUSIONS

The results of the study confirm that employing a hybrid ALNS + tabu search approach enables significant improvements in delivery scheduling efficiency under real-world operational conditions. The algorithm demonstrates high effectiveness not only in minimizing route lengths but also in reorganizing delivery structures in compliance with time and capacity constraints. The improvements in route quality were achieved without increasing the number of vehicles or reducing the number of service orders, which validates the approach's practical utility in day-to-day fleet management.

The proposed approach constitutes a valuable tool for both logistics researchers and practitioners, offering quantifiable benefits in terms of route shortening, reduced operational costs, and improved route structure, all while maintaining full service completion.

REFERENCES

1. Kucukoglu I, Dewil R, Cattrysse D. The electric vehicle routing problem and its variations: A literature review. Comput Ind Eng. 2021;161:107650.
2. Semenov I, Świderski A, Borucka A, Guzanek P. Concept of early prediction and identification of truck vehicle failures supported by in-vehicle telematics platform based on abnormality detection algorithm. Appl Sci. 2024;14(16). https://doi.org/10.3390/app14167191
3. Pham QD, Nguyen TH, Bui QT. Modeling and solving a multi-trip multi-distribution center vehicle routing problem with lower-bound capacity constraints. Comput Ind Eng. 2022;172:108597.
4. Salgado Duarte Y, Szpytko J. Reliability-oriented twin model for integrating offshore wind farm maintenance activities. Eksploatacja Niezawodn. 2024. https://doi.org/10.17531/ein/199355
5. Adamiak B, Andrych-Zalewska M, Merkisz J, Chłopek Z. The uniqueness of pollutant emission and fuel consumption test results for road vehicles tested on a chassis dynamometer. Eksploatacja Niezawodn. 2025;27(1). https://doi.org/10.17531/ein/195747
6. Sang T, Zhu K, Shen J, Yang L. An uncertain programming model for fixed charge transportation problem with item sampling rates. Eksploatacja Niezawodn. 2025;27(1). https://doi.org/10.17531/ein/192165
7. Lao J, Wang X, Zhang M, Chen Z, Guo J. Analysis and improvement of coal-loading performance and reliability of thin seam coal shearer drums. Eksploatacja Niezawodn. 2025;27(2). https://doi.org/10.17531/ein/194674
8. He X, Zhen L. Column-and-row generation based exact algorithm for relay-based on-demand delivery systems. Transp Res B Methodol. 2025;196:103223.
9. Liu R, Wang C, Ouyang H, Wu Z. Exact algorithm and machine learning-based heuristic for the stochastic lot streaming and scheduling problem. IISE Trans. 2025;57(4):408–22.
10. Kolankeh AK. A Hybrid Optimization algorithm for large-scale combinatorial problems in cloud computing environments. ALCOM J Algorithm Comput. 2025;1(01):1–12.
11. Tafakkori K, Tavakkoli-Moghaddam R, Siadat A. A data-driven sustainable scheduling model for dispatch-steerable last-mile delivery systems with negotiable time windows. Eng Appl Artif Intell. 2025;124:111280. https://doi.org/10.1016/j.engappai.2025.111280
12. Cengiz Toklu M. A fuzzy multi-criteria approach based on Clarke and Wright savings algorithm for vehicle routing problem in humanitarian aid distribution. J Intell Manuf. 2023;34(5):2241–61.
13. Özdağ R. A novel hybrid path planning method for sweep coverage of multiple UAVs. J Supercomput. 2025;81(1):83.
14. Sinaga RP, Marpaung F. Perbandingan algoritma cheapest insertion heuristic dan nearest neighbor dalam menyelesaikan traveling salesman problem. J Riset Rumpun Mat Ilmu Penget Alam. 2023;2(2):238–47.
15. Gao X, Cheng Z, Gao T, Wang R, Jiang K, Dong E. The optimal joint preventive maintenance strategy for the equipment under two-dimensional extended warranty. Eksploatacja Niezawodn. 2025. https://doi.org/10.17531/ein/203763
16. Zhu L. Energy management in microgrid integrated with ultracapacitor-equipped electric vehicles and renewable resources using hybrid algorithm perspective. Eksploatacja Niezawodn. 2025. https://doi.org/10.17531/ein/200713
17. Gao H, Wang Y, Xu J, Qin H. Fatigue strength reliability assessment of turbofan blades subjected to intake disturbances based on the improved kriging model. Eksploatacja Niezawodn. 2025;27(2). https://doi.org/10.17531/ein/194175

18. Zhang Y, Wang L. Real-time fault monitoring method for logistics vehicles based on chaotic ant colony algorithm. Eksploatacja Niezawodn. 2025. https://doi.org/10.17531/ein/203395
19. Rojek I, Jasiulewicz-Kaczmarek M, Piszcz A, Galas K, Mikołajewski D. Review of the 6G-based supply chain management within industry 4.0/5.0 Paradigm. Electronics. 2024;13(13):2624.
20. Merkisz J, Sordyl A, Chłopek Z. Non-repeatability of the WLTP vehicle test results. Arch Transp. 2024;71(3):25–49.
21. Zhu W, Cai W, Kong H. Optimal path planning based on ACO in intelligent transportation. Int J Cogn Comput Eng. 2025;6:100187. https://doi.org/10.1016/j.ijcce.2025.02.006
22. Hensel L, Grajewski V, Matteis T, Liedtke G. Modelling commercial traffic for a metropolitan region. Eur Transp Res Rev. 2025;17(1):717. https://doi.org/10.1186/s12544-025-00717-w
23. Danilevičius A, Danilevičienė I, Karpenko M, Stosiak M, Skačkauskas P. Determination of the instantaneous noise level using a discrete road traffic flow method. Promet – Traffic & Transportation. 2021; 33(1): 89–98. https://doi.org/10.7307/ptt.v37i1.597
24. Danilevičius A, Karpenko M, Křivánek V. Research on the noise pollution from different vehicle categories in the urban area. Transport. 2023;38(2):120–132. https://doi.org/10.3846/transport.2023.18666
25. Karpenko M, Prentkovskis O, Skačkauskas P. Analysing the impact of electric kick-scooters on drivers: vibration and frequency transmission during the ride on different types of urban pavements. Eksploatacja Niezawodn. 2025;27(2):199893. https://doi.org/10.17531/ein/199893