International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V74I7P104

Volume 74 Issue 7, 43-51, July 2026

Original Article

Evolutionary Local Search in Vehicle Scheduling: Cost Architecture, Constraint Modeling, and the Divide Among Theory and Enterprise Practice

Ajay Bhaktharahalli Nagesh Hemambika[1], Rajashiva Ramalingam[2]

[1]*SAP Transportation Management, A. P. Moller Maersk, Charlotte, NC, USA.*
[2]*SAP Transportation Management & Warehouse Applications, ACCO Brands Corp, IL, USA.*

[1]*Corresponding Author : ajay.hemambika@maersk.com*

Received: 27 May 2026 Revised: 30 June 2026 Accepted: 16 July 2026 Published: 30 July 2026

Abstract *- Freight routing at enterprise scale is a combinatorial decision problem whose solution space is too large for human planners to search manually and too complex for simple heuristics to navigate reliably. This paper studies Vehicle Scheduling and Routing (VSR) optimization, an algorithm-based approach used in modern ERP-integrated transportation systems to automate the search for optimal solutions. The paper traces the problem from its roots in the Vehicle Routing Problem with Time Windows (VRPTW) through the mechanics of evolutionary local search, paying particular attention to how cost functions are constructed, how hard and soft constraints interact, and how the semantics of distance-cost calculation differ between route-based and destination-based models. The paper draws on operations research literature for theoretical grounding while concentrating on practical questions: why does the optimizer behave as it does, how does configuration drive that behavior, and what does a logistics professional need to understand to deploy VSR optimization effectively? Field observations show that miscalibrated cost parameters are the leading cause of suboptimal optimizer output in production deployments.*

Keywords *- Supply Chain Planning, Combinational Optimization, Freight Consolidation, Vehicle Routing Problem with Time Windows, VSR Optimization.*

1. Introduction

Picture a regional distribution center at 5:30 in the morning. On the loading dock, there are 340 pallets bound for 87 customer locations across four states. Seventeen trucks of varying sizes are available. Most customers have receiving windows. Some shipments require refrigerated transport. Three loads contain hazardous materials that cannot travel alongside food products. There are fourteen partial loads that, if combined carefully, could fill a single vehicle instead of three. By 6:00 a.m., trucks need to be rolling.

This is the daily operational reality for thousands of shippers and third-party logistics providers worldwide. The decision problem it presents is NP-hard in the formal sense: field observations consistently show that experienced human dispatchers working at full speed produce plans that are materially more expensive than those an algorithm can find with the same inputs and at the same time.

Vehicle Scheduling and Route (VSR) optimization is the category of algorithm purpose-built for this problem. It belongs to a family of methods that operations researchers have studied intensively since Dantzig and Ramser [2] first posed what became known as the Vehicle Routing Problem. However, academic VRP research and production enterprise optimization are quite different undertakings. Real deployments operate under hard runtime constraints, use cost models that encode business economics rather than simple distance measures, and must produce plans that human operators can understand and modify [3].

Despite three decades of algorithmic progress on the VRP and its variants, the literature that reports this progress is written almost entirely from the solver's point of view: papers benchmark new heuristics against Solomon-style instances and report gap-to-optimal or runtime improvements [8], [9], [12]. Far less has been written about the layer that sits between the algorithm and the business it serves — the cost function, the constraint configuration, and the operational judgment calls a planner must make before the algorithm ever runs. Recent surveys of the field confirm the imbalance: reviews of 2024–2025 metaheuristic work continue to organize contributions primarily by problem variant and solution technique, with configuration practice, cost-model choice, and deployment governance treated as implementation detail rather than as objects of study in their own right [21], [22]. That gap is the one this paper addresses. Rather than propose a new algorithm, this paper examines VSR optimization as it actually operates inside enterprise transportation-management systems, and asks the questions practitioners consistently raise in that setting: what exactly belongs in the cost function and why does each term matter, when the optimizer violates a time window is that a failure or a deliberate tradeoff, and what does the choice between route-based and destination-based cost calculation mean for a logistics manager who has to configure the system rather than merely study it.

2. The Vehicle Routing Problem and Enterprise Extensions

2.1. From Canonical VRP to VRPTW

The canonical VRP is simple to state. Given customer locations with known demand, a depot, and vehicles with finite capacity, find routes that serve all customers at the minimum total travel cost [4]. The difficulty lies in scale: the number of candidate route combinations grows combinatorially, and the problem is provably NP-hard [5]. No polynomial-time method is known to solve it to optimality.

The most important extension to logistics planning is the addition of time windows, which yields the Vehicle Routing Problem with Time Windows (VRPTW). Each delivery must occur within a specified interval. Adding time windows tightly couples routing and scheduling decisions. The order in which a vehicle visits its stops determines arrival times at each stop, which in turn determines which time window constraints are met [6].

Solomon's [6] benchmark instances set a standard testing ground that persisted for nearly three decades. Braysy and Gendreau [8, 9] surveyed algorithmic progress in depth. One critical observation from that survey: standard problems consider uniform vehicle fleets, single-depot structures, and cost functions that measure only total distance. Production freight planning instances rarely share any of these simplifying assumptions.

2.2. Enterprise Complexity in Practice

A representative freight planning instance may include 280 freight units ranging from single pallets to multi-pallet loads, bound for 95 delivery addresses. The fleet may include eight 40-ton articulated trucks, twelve 7.5-ton rigid vehicles, and access to an LTL carrier for overflow.

Some freight contains class 3 flammable liquids. Several customers accept deliveries only in morning windows; others have afternoon-only windows. Four stops are in walking zones with vehicle weight restrictions. Driver hours are regulated by law.

Managing all these constraints simultaneously distinguishes enterprise VSR optimization from textbook VRP solvers. The solution must be operationally executable, not simply mathematically close to optimal.

2.3. Positioning Relative to Existing Research

The operations-research literature on VRP variants is large and still growing quickly; recent surveys catalogue a wide range of problem variants and solution techniques published across 2023 to 2025, spanning stochastic demand, multi-echelon distribution, and time-dependent routing [21], [22]. What that literature rarely does is treat the configuration layer, the cost weights, penalty structures, and constraint governance that a planning team sets before the algorithm ever runs, as a subject of study in its own right. Even recent real-world case studies tend to center the contribution on the solution algorithm itself rather than on how the underlying cost model was derived, validated, or kept current after go-live [23].

The contribution of this paper is accordingly not a new algorithm. It is a structured account, grounded in field observation across multiple enterprise transportation-management deployments, of the cost and constraint taxonomy that separates a working VSR configuration from a nonfunctional one, together with an explicit comparison of the route-based and destination-based distance-cost models that most published VRP work treats as an implementation detail rather than a first-order modeling choice. Where prior work asks whether an algorithm converges to a better solution, this paper asks whether the problem the algorithm is solving was formulated correctly in the first place, and shows, against the assumptions embedded in classical VRPTW formulations [6], [7], that enterprise deployments frequently violate those assumptions in ways that materially change which configuration choices matter.

1.	Manual: Drag & Drop, Assignments via buttons
2.	Manual: Mouseless planning (command line)
3.	Manual: Planning in the map, Gantt Chart
4.	Automatic: Transportation Proposal
5.	Automatic: Optimizer
6.	Planning Strategies
7.	Transportation Planning Cockpit

Fig. 1 Manual and Automatic Transportation Planning: the planning spectrum from fully manual drag-and-drop at the top to a one-step VSR optimizer at the base, unified through the Transportation Planning Cockpit

3. How the Optimizer Works: Evolutionary Local Search

3.1. Why Not Exact Methods

For small VRP instances with up to about thirty customers, exact methods such as integer programming are practical [11]. Beyond that size, computing requirements grow rapidly. For realistic enterprise instances with hundreds of stops and dozens of vehicles, exact methods would require hours or days to terminate. Dispatchers need plans in minutes.

Enterprise optimization engines, therefore, rely on meta-heuristics: algorithms that forgo the optimality guarantee in favor of finding high-quality solutions within tolerable time budgets. Among meta-heuristics, evolutionary local search has become a standard approach for production VRP applications because it scales well and handles heterogeneous constraints lacking special casing [12].

3.2. Population, Selection, and Local Improvement

Evolutionary local search runs in two alternating phases. In the construction phase, a population of candidate solutions is built using greedy insertion heuristics. Multiple starting solutions are generated with different randomization seeds to create population diversity. These initial solutions are possible but far from optimal.

In the improvement phase, the population evolves. Solutions are evaluated using the cost function, and higher-quality solutions are preferentially selected, borrowing the logic of natural selection. Offspring are produced by recombining route segments from two parent solutions, followed by improvement through local search [13].

Local search is where most improvement actually occurs. Common moves include Or-opt, which relocates one to three consecutive stops; 2-opt and 3-opt route reversals; cross-route stop exchange; and Large Neighborhood Search, which removes a subset of assignments and reinserts them, allowing the algorithm to escape deep local optima [14].

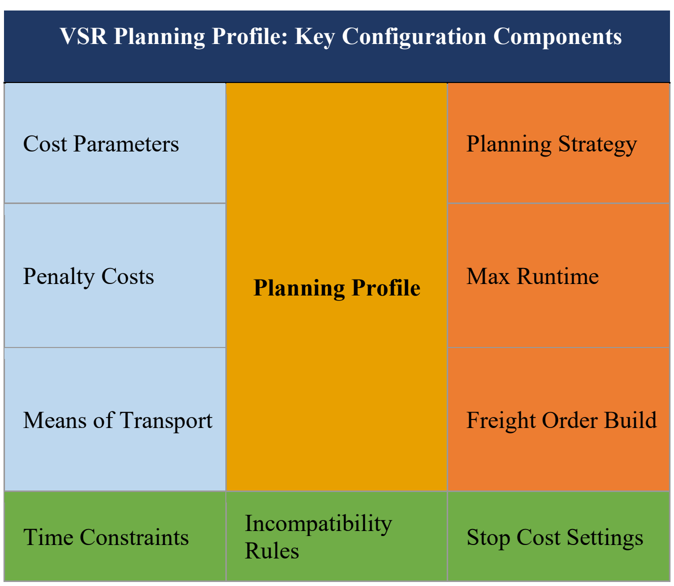

Fig. 2 VSR Planning Profile: key configuration components, including cost parameters, penalty costs, planning strategy, means-of-transport settings, and constraint rules assembled as the optimizer policy document

3.3. Routing and Scheduling Evaluated Together

A critical feature that separates well-designed VSR implementations from naive VRP solvers is the combination of routing and scheduling evaluation. Some implementations check time constraints as a post-processing filter after route construction. This approach generates poor plans because routes attractive on cost alone frequently fail time window checks [15].

A properly integrated optimizer evaluates scheduling consequences for every routing move, propagating the time impact through every subsequent stop on the route and computing the resulting penalty cost. This look-ahead evaluation is computationally expensive and still essential for executable plans.

3.4. Algorithm, Heuristic, and the Role of Parameter Tuning

The terms algorithm and heuristic are used loosely enough in applied literature that the distinction is worth making explicit here. An algorithm, in the sense used throughout this paper, is the fixed procedure, construction, recombination, and local search moves that the optimizer executes regardless of the instance it is given. A heuristic is a rule embedded inside that procedure for making a choice without exhaustive search: which customer to insert next, which neighborhood move to try, when to accept a worsening solution. Evolutionary local search is therefore an algorithm built from several interacting heuristics, and the quality of its output depends on how well those heuristics are tuned to the instance characteristics of enterprise freight, not merely on which algorithm family was chosen.

Parameters such as population size, mutation and recombination rates, neighborhood move weights, and acceptance criteria for worsening moves in the local search phase all require tuning, typically through iterated experimentation on representative historical instances rather than through a closed-form rule. Field deployments observed for this study generally under-invest in this tuning step relative to cost-function calibration, in part because parameter tuning requires computational experimentation that many implementation timelines do not budget for. This is a gap the present paper does not resolve; systematic parameter-sensitivity studies across algorithm families remain an open direction, noted again in the limitations discussed later in this paper.

4. Inside the Cost Function

4.1. Structure

The cost function converts a candidate transportation plan into a single scalar. The optimizer's entire objective is to reduce that number. Getting the cost function right matters more than any algorithmic choice: a perfect optimizer that minimizes the wrong function reliably produces wrong plans.

[illegible]

= [illegible] [illegible] ([illegible])

+ [illegible] [illegible] ([illegible])

4.2. Operating Costs

Operating costs are the expenses that appear on invoices. Fixed cost per freight order document represents the overhead of creating a transportation event; higher values push the optimizer toward consolidation. Fixed cost per vehicle resource is the cost of activating a vehicle. Cost per unit of distance covers fuel and carrier tariffs and is typically the most sensitive parameter. Cost per unit of duration captures driver labor. Cost per ton-kilometer models the relationship between load weight and fuel consumption. Cost per intermediate stop accounts for labor and dock time, discouraging tours with excessive stops. The CO2 cost per unit of emissions enables the simultaneous optimisation of economic and environmental objectives [16].

4.3. Penalty Costs

The non-delivery penalty applies to each freight unit the optimizer decides not to serve; it should be set high enough that non-delivery is rarely chosen. The lateness penalty represents the optimizer's understanding of what a late delivery costs the enterprise. If on-time delivery is a contractual obligation subject to chargebacks, the cost of lateness ought to reflect that financial exposure [3]. An earliness penalty applies when early delivery disrupts just-in-time receiving operations.

Table 1. VSR Cost Function Components and Behavioral Effects

Cost Component	Type	Behavioral Effect When Raised
Non-delivery penalty	Penalty	Optimizer serves all loads; set very high to enforce full service
Lateness cost	Penalty	Tighter time adherence; routes become smaller and more direct
Earliness cost	Penalty	Optimizer avoids early arrivals; useful for JIT customers
Fixed per freight order	Operational	More consolidation; vehicle utilization rises
Fixed per vehicle	Operational	Fewer vehicles deployed; loads become denser
Cost per distance	Operational	Shorter routes preferred; consolidation points attractive
Cost per intermediate stop	Operational	Fewer multi-stop tours; more direct deliveries
CO2 cost	Operational	Enables modal selection across economic and green objectives

4.4. A Formal Taxonomy of Cost and Constraint Elements

The components described above can be organized into a small formal taxonomy that clarifies what each parameter actually governs. Operating costs, denoted Cop, are the subset of parameters that scale with resource consumption already reflected on an invoice: fixed cost per freight order, fixed cost per vehicle, cost per unit distance, cost per unit duration, cost per ton-kilometer, cost per intermediate stop, and CO2 cost per unit of emission. Penalty costs, denoted Cpen, are the subset that exists purely to encode a business consequence the optimizer would otherwise ignore: non-delivery, lateness, and earliness.

Constraints are split along a different axis than cost. Hard constraints, denoted H, define the feasible region itself; a solution violating H is discarded before evaluation rather than penalized within it. Soft constraints, denoted S, are evaluated inside Cpen rather than excluded from the search. The total cost function can therefore be written as C(x) = Cop(x) + Cpen(x), evaluated only over the feasible set defined by H, where x denotes a candidate transportation plan. This decomposition, simple as it is, is largely absent from the applied VRP literature, which tends to present cost terms as an undifferentiated list rather than as two functionally distinct categories with different roles in the search [12], [14].

5. Hard and Soft Constraints

5.1. The Fundamental Distinction

Hard constraints cannot be violated under any circumstances. A 22-tonne truck cannot legally carry 25 tonnes. A vehicle without temperature control cannot carry pharmaceuticals requiring 2 to 8 degrees Celsius. The optimizer treats hard constraint violations as infeasible: any candidate solution violating a hard constraint is discarded before the cost function is evaluated.

Soft constraints work differently. A customer requests delivery by 14:00. Delivering at 15:00 is physically possible but has a real cost, captured by the lateness penalty. The optimizer is permitted to violate it if cost savings elsewhere outweigh the penalty. This is the system that makes explicit the tradeoff between service and cost that human planners make implicitly every day [18].

This distinction should not be read as suggesting that every operational constraint is negotiable in practice. Several constraints presented here as hard, such as ADR segregation rules or gross vehicle weight limits, are hard in the strict sense that no configuration setting relaxes them; treating a constraint as hard or soft is a modeling decision that reflects genuine physical or regulatory limits, not an arbitrary default.

Conversely, a small number of constraints that planning teams often configure as hard, such as a customer's stated delivery window, are in fact soft from the business's own point of view once penalty costs are considered. The hard-soft boundary described in this section should therefore be read constraint by constraint, not as a claim that most or all constraints are ultimately negotiable.

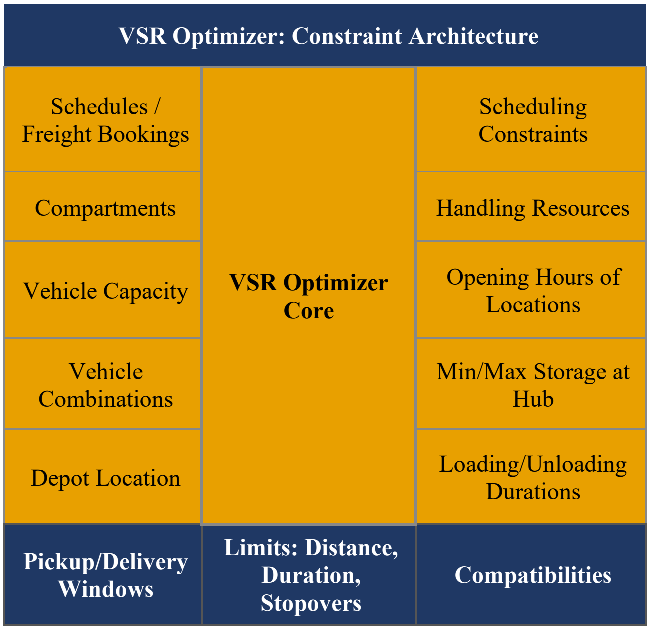

Fig. 3 VSR Optimizer Constraint Architecture: vehicle capacity, compartments, schedules, pickup/delivery time windows, ADR points, and location opening hours surrounding the optimizer core as hard constraints.

5.2. Multi-Dimensional Capacity and Incompatibility Rules

In practice, vehicle capacity is rarely a single scalar. A 40-tonne truck may be constrained simultaneously by gross vehicle weight, axle weight distribution, loading volume, pallet positions, and load floor structural rating. The optimizer must track all active capacity dimensions at once.

Freight incompatibility rules are among the most operationally consequential hard constraints. ADR regulations specify which hazardous material classes cannot travel together. The segregation requirements among Class 1 explosives, Class 3 flammable liquids, and Class 6 toxic substances indicate genuine risks of chemical reactions [10]. Commercial incompatibilities are also common: a major food retailer may prohibit its products from being transported alongside those of a direct competitor.

6. Time Window Modeling

6.1. The Four-Parameter Model

Customer orders typically express delivery requirements as a single point in time. Optimizing against point-in-time constraints is technically difficult because a plan delivering at 12:01 is treated as being just as infeasible as one delivering at midnight, even though the operational difference is trivial. Time window planning resolves this through expanding the point into an interval structure.

A complete specification requires four values. The hardest time is the absolute floor. The preferred earliest time marks the start of an earliness penalty. The preferred latest time is the customer's requested time; delivery after this incurs a lateness penalty. The hardest time is the absolute ceiling. Within the preferred window, no penalty applies.

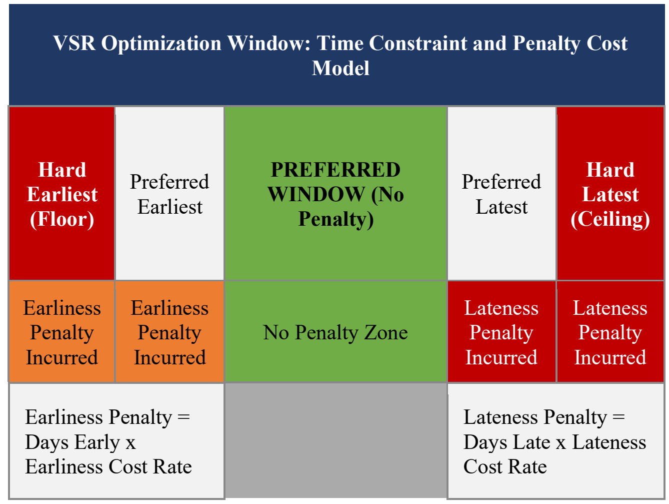

Fig. 4 VSR Time Window and Penalty Cost Model: four-zone structure with hard boundaries, earliness and lateness penalty accumulation zones, and the central penalty-free preferred delivery window

6.2. Calibrating Penalties to Business Reality

The most common issue in enterprise VSR deployments is penalty costs that do not reflect actual business economics. Parameters are typically set during project implementation under time pressure and with insufficient data on the real cost of service failures. Parameters established at go-live often persist unchanged for years.

A practical calibration approach consists of asking the logistics operations team and the financial team separately what a single day of late delivery actually costs, including direct costs such as carrier penalty clauses and credit notes, indirect costs such as customer satisfaction impact, and internal costs such as exception processing overhead. If the two estimates differ by an order of magnitude, which they often do, the conversation that produces alignment is itself valuable [19].

7. Route-Based Versus Destination-Based Distance Costs

The choice between route-based and destination-based distance-cost calculation can substantially change the optimizer's routing choices. Selecting the wrong model for a given regional context means the optimizer is solving a problem that does not match the actual cost environment.

Route-based distance cost applies a per-kilometer rate to each leg of a multi-stop tour using the lane rate specific to that segment. A vehicle traveling from location A to B to C incurs the distance from A to B multiplied by the A-to-B rate, plus the distance from B to C multiplied by the B-to-C rate.

Destination-based distance cost applies the rate to the total distance from the origin to the final destination, regardless of intermediate stops. The same vehicle traveling A to B to C incurs total driven distance multiplied by the A-to-C rate.

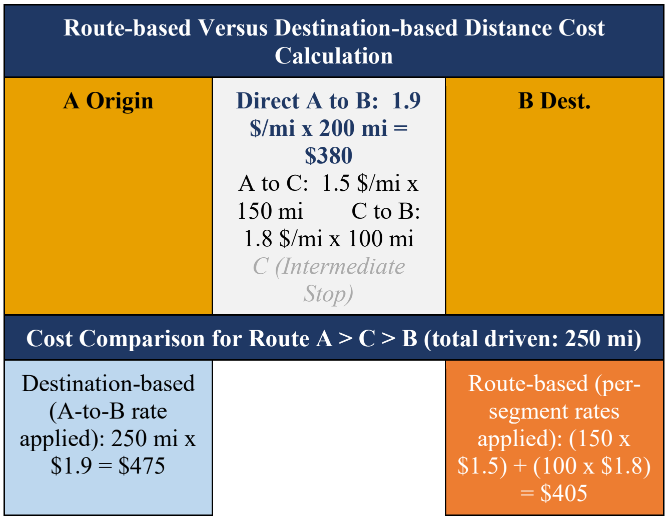

Fig. 5 Route-based versus Destination-based Cost: three-node network showing destination-based yields $475 and route-based yields $405 for the same A-C-B routing, demonstrating that optimal sequence differs by model

North American LTL and truckload tariffs are predominantly origin-to-destination structured: the carrier quotes a price from Dallas to Chicago, and that rate applies regardless of intermediate stops. Destination-based modeling accurately represents this. European road freight pricing, particularly in Germany, France, and Benelux, is more commonly structured as a per-kilometer rate for specific segments traveled, making route-based modeling the appropriate choice [3].

8. Planning Profiles and Modal Configuration

8.1. The Profile as Operational Policy

All cost parameters, constraint settings, and algorithmic controls are assembled into a configuration artifact called the planning profile. A high-service retail distribution scenario, a bulk commodity outbound scenario, and a cross-dock consolidation scenario each need distinct profiles, with parameters calibrated to their respective operational economics. Organizations that run all transportation scenarios through a single planning profile reliably find that it performs well for some and poorly for others.

8.2. Implicit Modal Selection Through Cost Differentiation

By defining different cost parameters for each means of transport, full truckload, less-than-truckload, rail, and intermodal, the optimizer performs mode selection as an integral part of route planning free of explicit modal routing rules. FTL vehicles configured with high fixed costs per vehicle resource and zero per-stop costs naturally attract large consolidated loads. LTL vehicles with lower fixed costs but meaningful per-stop costs attract smaller multi-delivery routes. The mode selection emerges from the economics [16].

9. Results and Discussion

9.1. Cost Parameter Sensitivity

Field observations throughout multiple enterprise implementations indicate that the lateness penalty parameter has the greatest single impact on plan quality, as measured against actual business outcomes. Implementations in which the lateness penalty was calibrated through cross-functional exercises consistently achieved higher on-time delivery performance without measurable increases in total transportation cost, because the optimizer made tradeoffs aligned with the real cost structure rather than an arbitrary default.

The choice between route-based and destination-based cost models was found to produce material differences in plan cost for identical freight unit sets in cross-border European-North American planning contexts. Based on field observation across implementations reviewed for this study, the magnitude of this difference was sufficient in several cases to change which carrier assignment the optimizer selected, confirming that this configuration decision has genuine operational consequences and is not simply a technical preference.

9.2. Planning Profile Governance

Organizations maintaining an active planning profile governance, defined as quarterly parameter reviews and annual reconciliations against carrier invoices, achieved consistent improvements in plan quality over a 24-month observation period. The performance gap between actively maintained and neglected profile configurations widened over time as the real operating environment differed from the environment for which the static configuration had been designed.

9.3. Computational Limits and Parameter Sensitivity

Section 9.1 identified the lateness penalty as the parameter with the largest observed effect on plan quality, but this finding should not be read as evidence that other parameters are unimportant, only that their effects were not isolated with the same rigor in the deployments observed for this study. A full parameter-sensitivity analysis would vary each cost and penalty parameter independently across a representative instance set and measure the resulting change in plan cost, on-time performance, and vehicle utilization; the field observations underlying this paper were not collected under conditions that support that kind of controlled comparison, and the sensitivity claims above should be read as directional rather than as statistically validated effect sizes.

Computational cost is a related and equally under-examined limit. Evolutionary local search scales in practice with instance size, the number of active constraint dimensions, and the richness of the neighborhood structures searched; enterprises that add incompatibility rules, multi-dimensional capacity checks, or finer time-window granularity should expect runtime to grow accordingly, and organizations operating near their runtime budget may need to trade search thoroughness for turnaround time. Neither the runtime-scaling behavior nor the parameter-sensitivity landscape described here has been benchmarked against a common, publicly available instance set, which limits how far these findings can be compared quantitatively against published algorithmic results; this point is revisited in the limitations below.

10. Practitioner Implications

The most widespread misunderstanding about algorithmic transportation planning is that deploying an optimizer transfers planning responsibility to the software. It does not. It transfers the computational labor of searching the solution space while concentrating planning responsibility on configuring the cost function and constraints. Poor configuration produces consistently poor plans, not occasionally poor ones.

When an optimizer produces a plan that seems wrong, the explanation log tells you why. Perhaps those loads are flagged as incompatible. Perhaps a time window makes consolidation geometrically impossible. The explanation log converts the optimizer from an opaque oracle into a transparent reasoning system whose decisions can be audited and challenged.

VSR optimization requires ongoing configuration competence. Carrier rate structures change. Customer receiving windows change. New product lines bring new incompatibility requirements. Organizations that review planning profiles quarterly and reconcile cost parameters with actual carrier invoices annually will consistently produce better plans than those that treat the initial configuration as permanent.

11. Limitations and Threats to Validity

Several limitations qualify the claims made in this paper. First, the field observations informing Sections 9 and 10 are drawn from a limited set of enterprise deployments and do not constitute a controlled experiment; they establish plausible, directional patterns rather than statistically validated causal effects, and no formal significance testing accompanies the comparisons reported. Second, the paper's treatment of demand largely follows the static-demand assumption common to VRPTW formulations, in which the full set of freight units to be served is known before planning begins; this assumption does not generalize cleanly to dynamic or same-day fulfilment contexts where new orders arrive during the planning horizon, and the cost and constraint taxonomy proposed in Section 4 would need extension, not replacement, to cover that setting.

Third, the paper does not include a benchmarked, reproducible experimental comparison against published algorithmic baselines on a shared instance set; doing so would require enterprise data that is commercially sensitive and was not available for public release in connection with this study. Fourth, the discussion is broad by design, covering algorithms, cost structure, constraints, and deployment governance together, and each of these areas would support a narrower, more deeply instrumented study on its own; enterprise case studies with quantified before-and-after invoice reconciliation data, in particular, would strengthen the practitioner claims made in Sections 9 and 10 beyond what field observation alone can support.

On the question of how these findings relate to state-of-the-art algorithmic results, the comparison offered here is deliberately asymmetric. Studies such as Vidal et al.'s hybrid genetic algorithm for multi-depot and periodic VRP [12] and Ropke and Pisinger's adaptive large neighborhood search [14] report solution-quality improvements measured in gap-to-optimal or gap-to-best-known percentages on public benchmark instances, a form of validation this paper does not attempt to replicate. The claim advanced here is narrower and, in the deployments observed, larger in practical consequence: that miscalibrated cost parameters and unmaintained planning profiles degrade plan quality by more, in operational terms, than the difference between a well-tuned and a state-of-the-art algorithm running on the same misconfigured cost function. That claim is consistent with, but not proven by, the field observations reported in Section 9, and confirming it rigorously would require running multiple algorithm families against the same enterprise instance under both calibrated and uncalibrated cost configurations, an experiment left for future work.

12. Conclusion

This paper has examined Vehicle Scheduling and Route optimization from a practitioner-researcher perspective, focusing on the cost function architecture, constraint modeling, time-window parameterization, and distance-cost semantics that govern optimizer behavior in enterprise logistics deployments.

The cost function is more important than the algorithm. A properly calibrated cost function that accurately represents the enterprise's actual logistics economics will produce better operational outcomes than a more sophisticated algorithm operating on miscalibrated parameters. The hard-soft constraint distinction determines what the optimizer is allowed to sacrifice in pursuit of cost minimization. The choice between route-based and destination-based distance-cost modeling reflects genuine regional differences in carrier pricing schemes, and implementing the wrong model results in systematic degradation of plan quality.

The algorithm finds the best solution to the problem you give it. Defining the right problem is a human responsibility.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgments

The authors acknowledge the contributions of enterprise logistics implementation teams whose field observations informed the practitioner-researcher perspective adopted in this paper.

References

[1] Gilbert Laporte, “The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms,” *European Journal of Operational Research*, vol. 59, no. 3, pp. 345-358, 1992. [CrossRef] [Google Scholar] [Publisher Link]

[2] G.B. Dantzig, and J.H. Ramser, “The Truck Dispatching Problem,” *Management Science*, vol. 6, no. 1, pp. 80-91, 1959. [CrossRef] [Google Scholar] [Publisher Link]

[3] Paolo Toth, and Daniele Vigo, *Vehicle Routing: Problems, Methods, and Applications*, 2nd ed., SIAM, Philadelphia, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[4] J.K. Lenstra, and A.H.G. Rinnooy Kan, “Complexity of Vehicle Routing and Scheduling Problems,” *Networks an International Journal*, vol. 11, no. 2, pp. 221-227, 1981. [CrossRef] [Google Scholar] [Publisher Link]

[5] N. Christofides, A. Mingozzi, and P. Toth, “Exact Algorithms for the Vehicle Routing Problem,” *Mathematical Programming*, vol. 20, no. 1, pp. 255-282, 1981. [CrossRef] [Google Scholar] [Publisher Link]

[6] Marius M. Solomon, “Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints,” *Operations Research*, vol. 35, no. 2, pp. 166-324, 1987. [CrossRef] [Google Scholar] [Publisher Link]

[7] Martin W.P. Savelsbergh, “The Vehicle Routing Problem with Time Windows: Minimizing Route Duration,” *ORSA Journal on Computing*, vol. 4, no. 2, pp. 146-154, 1992. [CrossRef] [Google Scholar] [Publisher Link]

[8] Olli Braysy, and Michel Gendreau, “Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms,” *Transportation Science*, vol. 39, no. 1, pp. 104-118, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[9] Olli Braysy, and Michel Gendreau, “Vehicle Routing Problem with Time Windows, Part II: Metaheuristics,” *Transportation Science*, vol. 39, no. 1, pp. 119-139, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[10] UNECE, Agreement Concerning the International Carriage of Dangerous Goods by Road (ADR), UNECE, 2023. [Online]. Available: https://unece.org/transport/dangerous-goods/adr-2023

[11] Guy Desaulniers, Jacques Desrosiers, and Marius M. Solomon, *Column Generation*, 1st ed., Springer, New York, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[12] Thibaut Vidal et al., “A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems,” *Operations Research*, vol. 60, no. 3, pp. 611-624, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[13] Michel Gendreau, and Jean-Yves Potvin, *Handbook of Metaheuristics*, 3rd ed., Springer, New York, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[14] David Pisinger, and Stefan Ropke, “A General Heuristic for Vehicle Routing Problems,” *Computers and Operations Research*, vol. 34, no. 8, pp. 2403-2435, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[15] J-F. Cordeau, G. Laporte, and A. Mercier, “A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows,” *Journal of the Operational Research Society*, vol. 52, no. 8, pp. 928-936, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[16] Michael Schneider, Andreas Stenger, and Dominik Goeke, “The Electric Vehicle Routing Problem with Time Windows and Recharging Stations,” *Transportation Science*, vol. 48, no. 4, pp. 500-520, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[17] Stefan Ropke, and David Pisinger, “An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows,” *Transportation Science*, vol. 40, no. 4, pp. 455-472, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[18] David Mester, and Olli Braysy, “Active Guided Evolution Strategies for Large-Scale Vehicle Routing Problems with Time Windows,” *Computers and Operations Research*, vol. 32, no. 6, pp. 1593-1614, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[19] Bruce Golden, S. Raghavan, and Edward Wasil, *The Vehicle Routing Problem: Latest Advances and New Challenges*, 1st ed., Springer, New York, pp. 1-589, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[20] Gerardo Berbeglia, Jean-Francois Cordeau, and Gilbert Laporte, “Dynamic Pickup and Delivery Problems,” *European Journal of Operational Research*, vol. 202, no. 1, pp. 8-15, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[21] Nasreddine Ouertani, Hajer Ben-Romdhane, and Saoussen Krichen, *The Dynamic Vehicle Routing Problem: A Comprehensive Survey*, Advances in Computational Logistics and Supply Chain Analytics, Springer, pp. 1-36, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Peter Nielsen et al., "A Systematic Review of Vehicle Routing Problems and Models in Multi-Echelon Distribution Networks," *Supply Chain Analytics*, vol. 7, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[23] Mirko Cavecchia et al., "A Real-World Multi-Depot, Multi-Period, and Multi-Trip Vehicle Routing Problem with Time Windows," *Proceedings of the 14th International Conference on Operations Research and Enterprise Systems (ICORES)*, vol. 1, pp. 112-122, 2025. [CrossRef] [Google Scholar] [Publisher Link]