State of the Art Algorithms for the Periodic Capacitated Vehicle Routing Problem: A Comprehensive Review

1. Introduction: The Strategic Imperative of Periodic Routing

The Periodic Capacitated Vehicle Routing Problem (PCVRP) represents one of the most intellectually demanding and industrially significant challenges in the domain of Operations Research (OR) and combinatorial optimization. While the classical Vehicle Routing Problem (VRP)—first introduced by Dantzig and Ramser in 1959—addresses the tactical optimization of routes for a single day, the PCVRP elevates this challenge to a strategic and tactical planning horizon.[1] In the PCVRP, the optimization scope expands from a snapshot in time to a multi-period continuum, typically spanning days or weeks, where customers require repeated service according to specific frequency requirements and permissible visit combinations.[4]

The transition from the standard Capacitated VRP (CVRP) to the Periodic variant is not merely an additive increase in variables; it fundamentally alters the complexity class and the structural nature of the problem. The PCVRP introduces a bi-level decision hierarchy: the *tactical* assignment of service patterns (specific combinations of visit days) to customers, and the *operational* construction of efficient vehicle routes for each day of the planning horizon based on those assignments.[3] This interdependence creates a combinatorial explosion where a suboptimal decision at the pattern assignment level can render the routing phase inefficient or infeasible, necessitating algorithms that can simultaneously explore both search spaces—assignment and routing—in a unified manner.[6]

1.1 Industrial Relevance and Applications

The practical ubiquity of the PCVRP cannot be overstated. It models the core operational logic of vast sectors of the global economy. In waste management, arguably the most direct application, municipalities do not optimize collection routes daily from scratch; rather, they assign households to specific collection days (e.g., Mondays and Thursdays) to balance workload and minimize fleet usage over a week.[3] Similarly, in the beverage and food distribution industry, retailers differ in their sales velocity, requiring replenishment at varying frequencies—high-volume supermarkets may need daily service, while smaller convenience stores require only weekly visits.[3]

Furthermore, the rise of "Service Choice" models in logistics has reinvigorated interest in PCVRP. Modern logistics providers often offer tiered service levels where customers can pay premiums for specific delivery windows or frequencies, transforming the constraints of the PCVRP into decision variables that impact revenue management.[8] The problem also extends to preventative maintenance, where technicians must visit equipment at regular intervals, and to the replenishment of vending machines, where the frequency of visits is dictated by consumption rates and stock capacities.[9] The ability to solve the PCVRP efficiently translates directly to reduced fuel consumption, smaller fleet requirements, and improved service consistency, identifying it as a critical lever for sustainable "Green Logistics".[10]

1.2 The Evolution of Algorithmic Approaches

The algorithmic landscape for solving the PCVRP has undergone a radical transformation over the past decade, accelerating significantly in the 2024-2025 period. Historically, the field was bifurcated into exact methods—capable of solving only small instances to optimality—and heuristic methods that provided feasible but often suboptimal solutions for larger, real-world instances.[4] However, the boundary between these approaches is blurring.

The development of **Branch-Cut-and-Price (BCP)** algorithms, exemplified by the **VRPSolver** framework, has pushed the limits of exact solvability to instances with over 100 customers, utilizing sophisticated techniques like ng-route relaxation and subset-row inequalities.[13] Simultaneously, the metaheuristic dominance of **Hybrid Genetic Search (HGS)** and **Adaptive Large Neighborhood Search (ALNS)** has been solidified by their integration into high-performance open-source libraries like **PyVRP**.[15]

Most notably, the emergence of **Neural Combinatorial Optimization (NCO)** has introduced a third paradigm. Deep Reinforcement Learning (DRL) and Large Language Model (LLM)-driven heuristic discovery are now challenging handcrafted algorithms. Innovations such as **Neural Deconstruction Search (NDS)** and **VRPAgent** have demonstrated the ability to learn problem-specific heuristics that outperform traditional methods on massive-scale instances, marking a pivotal moment where Artificial Intelligence begins to design the algorithms that drive logistics.[17]

This report provides an exhaustive analysis of these state-of-the-art methodologies. It dissects the mathematical underpinnings, operational mechanics, and comparative performance of current algorithms, offering a definitive reference for the landscape of PCVRP optimization in 2025.

---

2. Mathematical Formulations: The Structural Foundation

To comprehend the sophistication of modern algorithms, one must first analyze the mathematical scaffolding upon which they are built. The PCVRP is defined on a graph $G = (V, E)$, where $V = \{0, 1,..., n\}$ represents the set of nodes (with 0 being the depot and $V_c = \{1,..., n\}$ being the customers), and E is the set of edges connecting them. The planning horizon consists of T periods (days).

2.1 The Challenge of Pattern Selection

The defining characteristic of the PCVRP is the *visit pattern*. Each customer i has a service frequency f_i (number of visits required) and a set of allowable visit combinations R_i (patterns). For example, if $T=5$ (Mon-Fri) and a customer requires $f_i=2$, valid patterns might be $\{ \text{Mon, Thu} \}$ or $\{ \text{Tue, Fri} \}$. The solver must select exactly one pattern $r \in R_i$ for each customer.[9]

This selection process introduces variables y_{ir}, a binary variable equal to 1 if customer i is assigned pattern r. This assignment dictates the demand on specific days. If $y_{ir} = 1$ and day $t \in r$, then customer i effectively becomes a node in the CVRP subproblem for day t.[3] This structure naturally suggests a decomposition where the "Master Problem" handles pattern assignment and the "Subproblems" solve the resulting daily routings.

2.2 The Three-Index Flow Formulation

The most intuitive mathematical representation is the Three-Index Flow formulation, which extends the classical CVRP model. Let x_{ijkt} be a binary variable equal to 1 if vehicle k travels from node i to node j on day t.

The objective is to minimize the total travel cost:

$$\text{Minimize} \sum_{t=1}^{T} \sum_{k \in K} \sum_{(i,j) \in E} c_{ij} x_{ijkt}$$

Subject to:

1. Pattern Assignment Constraint: Each customer must be assigned exactly one valid pattern.

   $$\sum_{r \in R_i} y_{ir} = 1, \quad \forall i \in V_c$$
2. Service Consistency: If a customer is assigned a pattern, they must be visited on the days dictated by that pattern.

   $$\sum_{k \in K} \sum_{j \in V} x_{ijkt} = \sum_{r \in R_i, t \in r} y_{ir}, \quad \forall i \in V_c, \forall t \in \{1,..., T\}$$
3. **Flow Conservation and Capacity:** Standard flow balance constraints ensure vehicles enter and leave nodes, and that the sum of demands on any route does not exceed vehicle capacity Q.[19]

While semantically clear, this formulation is notoriously weak for computational purposes. The linear relaxation provides poor lower bounds, and the number of variables x_{ijkt} grows cubically with the problem size ($N \times N \times K \times T$). Consequently, this model is rarely used in state-of-the-art exact solvers, which favor path-based or set-partitioning formulations.[1]

2.3 Set Partitioning Formulation

State-of-the-art exact algorithms, particularly Branch-Cut-and-Price methods, rely on the Set Partitioning (SP) formulation. This approach creates a variable for every feasible route rather than every edge.

Let Ω_t be the set of all feasible routes for day t. Let λ_{lt} be a binary variable equal to 1 if route $l \in \Omega_t$ is selected. Let a_{ijl} be a binary parameter indicating if customer i is visited in route l.

The Master Problem (MP) minimizes the cost of selected routes:

$$\text{Minimize} \sum_{t=1}^{T} \sum_{l \in \Omega_t} c_l \lambda_{lt}$$

Subject to constraints ensuring that for each customer i, the total number of visits across all days equals the required frequency f_i, and that the visits conform to a valid pattern combination.[4]

The strength of this formulation lies in its handling of the routing constraints (capacity, time windows) *implicitly* within the definition of a "feasible route." This allows the complex constraints to be handled by the subproblem (pricing problem) rather than the master problem linear program. However, the number of feasible routes is exponential, necessitating Column Generation—generating variables λ_{lt} only as needed.[13]

2.4 Symmetry and Periodicity

A unique mathematical challenge in PCVRP is "Period Symmetry." If the fleet and demand are identical across all days, the days are essentially interchangeable. A solution that schedules a customer on Monday-Thursday might be mathematically identical in cost to one scheduling them on Tuesday-Friday. This symmetry creates a massive search tree in Branch-and-Bound algorithms, as the solver wastes time exploring symmetric duplicates.[9] Modern formulations incorporate symmetry-breaking constraints or aggregation techniques to mitigate this, forcing a canonical ordering of days or patterns to prune the search space.[20]

---

3. Exact Algorithms: The Pinnacle of Rigor

While heuristics are necessary for large-scale operations, exact algorithms provide the gold standard for solution quality. They guarantee optimality, providing a baseline against which all other methods are measured. The state-of-the-art for exact PCVRP solving is undeniably the

Branch-Cut-and-Price (BCP) framework, with **VRPSolver** representing the leading implementation.

3.1 Branch-Cut-and-Price (BCP) Framework

BCP is a hybrid methodology that combines three powerful OR techniques:

1. **Branch-and-Bound:** The overall tree search management.
2. **Column Generation (Branch-and-Price):** Handling the exponential number of variables in the Set Partitioning formulation. Instead of enumerating all routes, the algorithm solves a "Pricing Problem"—typically a Shortest Path Problem with Resource Constraints (SPPRC)—to find new routes with negative reduced costs to add to the LP relaxation.[13]
3. **Cutting Planes (Branch-and-Cut):** Strengthening the lower bound of the LP relaxation by adding valid inequalities (cuts) that slice off fractional solutions without removing integer feasible ones.[21]

In the context of PCVRP, the BCP approach is particularly potent because it decouples the complex routing constraints (handled in the pricing problem) from the pattern assignment constraints (handled in the master problem).

3.2 The Pricing Problem and ng-route Relaxation

The bottleneck of BCP is the pricing problem. Solving the elementary SPPRC (finding the shortest path visiting each node at most once) is NP-hard. To accelerate this, modern solvers utilize the **ng-route relaxation**.[13]

- *Mechanism:* The ng-route relaxation allows cycles (visiting a node more than once) but restricts them in a localized neighborhood. Each node has a defined "memory" of forbidden predecessors.
- *Impact:* This relaxation makes the pricing problem pseudo-polynomial, solvable via dynamic labeling algorithms. It provides a tight approximation of the true elementary shortest path, significantly speeding up column generation without excessively loosening the lower bound.[14]

3.3 Valid Inequalities: The Cutting Edge

The "Cut" component of BCP has seen significant innovation. Standard capacity cuts are insufficient for the tight bounds required in PCVRP.

- **Subset-Row Inequalities (SRIs):** These cuts operate on the master problem variables. They enforce that for any subset of rows (customers), the number of routes covering them respects the packing constraints. SRIs are notoriously difficult to separate but provide massive improvements in bound quality for 3-row subsets.[22]
- **Rank-1 Cuts:** These are a broader class of cuts that can be applied to the set partitioning polytope.
- **Pattern Compatibility Cuts:** Specific to PCVRP, these cuts link the pattern assignment

variables y_{ir} with the route variables λ_{lt}. They enforce logical consistency—e.g., "if customer i is not assigned a pattern containing Monday, no Monday route can visit customer i".[9]

3.4 VRPSolver: The Generic Exact Solver

The most significant software advancement in exact VRP solving is **VRPSolver**, developed by Pessoa, Sadykov, Uchoa, and Vanderbeck.

- *Unified Architecture:* VRPSolver is not just a solver but a mapping framework. It maps various VRP variants (CVRP, VRPTW, PCVRP) into a generic structure that the solver engine understands.
- *Performance:* VRPSolver holds the record for the largest PCVRP instances solved to optimality. It has closed instances with over 100 customers that remained open for decades. Its success is attributed to the integration of automatic dual price smoothing (stabilization), heuristic strong branching, and the sophisticated management of the bucket graph for labeling algorithms.[14]
- *Limitation:* Despite its power, the "curse of dimensionality" in the pattern space limits VRPSolver. For PCVRP, the addition of the time dimension means that an instance with 100 customers over 5 days behaves similarly to a static instance with 500 nodes in terms of variable complexity, limiting exact solvability to relatively small/medium instances.[24]

---

4. Metaheuristics: The Industrial Workhorses

For real-world logistics involving hundreds or thousands of customers, exact methods are computationally prohibitive. Metaheuristics, which trade guaranteed optimality for speed and scalability, are the standard. The current state-of-the-art is dominated by **Hybrid Genetic Search (HGS)**, **Slack Induction by String Removals (SISR)**, and **Adaptive Large Neighborhood Search (ALNS)**.

4.1 Hybrid Genetic Search (HGS)

HGS, particularly the variants developed by Vidal, Crainic, Gendreau, and Prins, is arguably the most successful VRP metaheuristic of the modern era. It combines the diversification of evolutionary algorithms with the intensification of local search.[5]

4.1.1 The "Split" Algorithm and Giant Tour Representation

The core innovation of HGS is its solution representation. Instead of encoding complex routes directly (which makes crossover difficult), HGS encodes a solution as a "Giant Tour"—a simple permutation of all customer visits.

- *Mechanism:* The **Split** algorithm (based on Prins, 2004) converts this permutation into an optimal set of feasible routes. It models the giant tour as a Directed Acyclic Graph (DAG) where arcs represent feasible routes extracted from the sequence. The shortest path in this DAG corresponds to the optimal segmentation of the giant tour into vehicle routes.[5]

- *PCVRP Adaptation:* For the Periodic problem, the representation is augmented. A chromosome consists of (1) the Pattern assignment for each customer, and (2) the Giant Tour permutation. The Split algorithm is applied *independently* for each day of the period. This decoupling allows the algorithm to explore the routing space and the pattern assignment space simultaneously.[6]

4.1.2 Adaptive Diversity Control

Premature convergence is the nemesis of Genetic Algorithms. HGS combats this with **Adaptive Diversity Control**.

- *Mechanism:* The fitness of an individual in the population is not determined solely by its cost but by a weighted sum of Cost and "Diversity Contribution" (distance from other individuals).
- *Subpopulations:* HGS maintains two subpopulations: Feasible and Infeasible. The Infeasible subpopulation allows routes that violate capacity constraints (with a penalty). This is crucial for navigating the disjoint solution landscape of the PCVRP, allowing the search to "tunnel" through infeasible regions to reach high-quality feasible basins.[5]

4.1.3 Performance Dominance

HGS consistently discovers Best Known Solutions (BKS) for the classical Cordeau and Golden benchmark instances. It is the engine behind the high-performance **PyVRP** library, making it the most accessible SOTA algorithm for practitioners.[15]

4.2 Slack Induction by String Removals (SISR)

While HGS relies on complex population dynamics, SISR (Christiaens & Vanden Berghe, 2020) champions simplicity. It is a "Ruin and Recreate" heuristic that has redefined the efficiency frontier.[27]

4.2.1 Spatial Slack and String Removal

SISR is built on the concept of "Spatial Slack"—the ability to insert a customer into a route without significant detour.

- *Ruin Strategy:* Traditional heuristics remove random customers. SISR removes "strings"—sequences of geographically adjacent customers. Removing a string creates a large, contiguous gap in the route, maximizing the spatial slack available for re-insertion.
- *Why String Removal?* Removing scattered nodes leaves "Swiss cheese" routes—full of small holes that are hard to fill efficiently. Removing a chain of nodes clears a whole region, allowing the "Recreate" phase to fundamentally restructure the local geography of the solution.[27]

4.2.2 Greedy Insertion with Blinks

The "Recreate" phase uses a greedy insertion heuristic. However, checking every possible insertion position for every customer is slow ($O(N^2)$). SISR introduces "Blinks."

- *Mechanism:* When evaluating insertion positions, the algorithm skips (blinks) a percentage of checks based on a probability parameter.
- *Impact:* This drastically reduces computational effort while introducing beneficial noise that prevents the greedy heuristic from getting stuck in local optima. For the PCVRP, where an insertion must be checked against multiple pattern possibilities on multiple days, the speedup from Blinks is exponential, allowing SISR to run millions of iterations in the time HGS runs thousands.[29]

4.3 Adaptive Large Neighborhood Search (ALNS)

ALNS remains a robust contender, particularly for "Rich" PCVRP variants involving heterogeneous fleets or complex side constraints where the HGS "Split" procedure becomes difficult to implement.[30]

- *Operator Portfolio:* ALNS maintains a portfolio of Destroy (Ruin) and Repair operators.
  - *Shaw Removal:* Removes customers that are related (by distance, demand, or pattern).
  - *Pattern Removal:* Specifically targets the pattern assignments, unassigning customers from their current days to force a re-evaluation of the tactical schedule.
  - *Regret Insertion:* Re-inserts customers based on the "regret" value—the difference in cost between their best and second-best insertion options. This prioritizes "difficult" customers.[31]
- *Adaptive Layer:* The algorithm learns which operators are successful during the search and biases the selection probability towards them. This allows ALNS to auto-tune itself to the specific characteristics of the instance (e.g., favoring pattern-swapping operators in instances with tight frequency constraints).[30]

---

5. Neural Combinatorial Optimization: The New Frontier (2024-2025)

The most transformative development in recent years is the application of Deep Learning to VRPs. Neural Combinatorial Optimization (NCO) has moved from "interesting academic curiosity" to "SOTA competitor," particularly through hybrid approaches.

5.1 Deep Reinforcement Learning (DRL) Approaches

Early NCO methods used "Constructive" DRL (e.g., Pointer Networks) to build solutions node-by-node. While fast, these struggled to match the quality of HGS. The current SOTA focuses on **Improvement Learning** and **Dynamic VRPs**.

- **Learning to Improve:** Instead of building a route from scratch, DRL agents are trained to

perform local search moves. The policy network takes a solution state and outputs a probability distribution over possible 2-opt or swap moves. This effectively replaces the handcrafted logic of a local search operator with a learned intuition that recognizes complex topological patterns.[32]
- **Dynamic Attention Networks:** For *Dynamic* PCVRPs—where orders arrive in real-time or demand is stochastic—DRL is superior to traditional heuristics. A "Dynamic Attention Network" encodes the evolving state of the graph (vehicle locations, revealed demands) into a latent vector. The decoder then outputs dispatching decisions that anticipate future demand clusters, a capability that myopic heuristics like Nearest Neighbor lack.[34]

5.2 Neural Deconstruction Search (NDS)

Neural Deconstruction Search (NDS) (2025) is a hybrid breakthrough that outperforms pure OR and pure DL methods.[18]

- *Concept:* NDS integrates a Deep Neural Network (DNN) into the LNS framework. It replaces the random or heuristic "Destroy" operators with a learned policy.
- *Mechanism:* The DNN is trained to identify "structural bottlenecks"—customers whose removal would yield the highest potential for improvement. It learns the "topology of bad decisions."
- *Hybrid Vigor:* By coupling this intelligent deconstruction with a standard, highly optimized greedy reconstruction, NDS leverages the best of both worlds: the pattern-recognition of AI and the constraint-handling precision of classical OR. Benchmarks show NDS finding better solutions in fewer iterations than SISR on medium-sized instances.[35]

5.3 VRPAgent and LLM-Driven Discovery

Perhaps the most radical innovation is **VRPAgent** (2025), which utilizes Large Language Models (LLMs) to *invent* heuristics.[17]

- *Evolutionary Prompting:* The framework treats code as genetic material. It prompts an LLM (e.g., GPT-4) to write a Python function for a local search operator. It evaluates this code on small instances, measures performance, and feeds the results back to the LLM to "mutate" and improve the code.
- *Discovery of Novel Operators:* VRPAgent has discovered heuristic operators that human experts had not conceived—complex, non-intuitive logic for swapping patterns and nodes.
- *SOTA Performance:* On large-scale PCVRP instances ($N > 1000$), heuristics discovered by VRPAgent have achieved negative optimality gaps relative to SISR (i.e., finding better solutions). This marks a paradigm shift: we are moving from *designing* algorithms to *managing* the AI agents that design them.[36]

---

6. Computational Benchmarking and Performance

Comparing algorithms requires rigorous benchmarking on standardized datasets. The primary benchmarks for PCVRP are the **Cordeau et al.** instances and the **Golden et al.** instances (adapted for periodicity).

6.1 Benchmark Metrics

Performance is typically measured by:

1. **Gap to Best Known Solution (BKS):** Percentage deviation from the best solution ever found.
2. **Runtime:** Wall-clock time to reach a certain quality.
3. **Scalability:** How performance degrades as N increases.

6.2 Comparative Performance Table

The following table synthesizes the performance landscape based on 2024-2025 literature data [18]:

Algorithm Class	Method	Optimal Range (N)	Gap to BKS	Key Strength	Key Weakness
Exact	**VRPSolver**	$N < 150$	0.0% (Optimal)	Proves optimality; handles rich constraints well via BCP mapping.	Exponential runtime scaling; memory limits on large pattern sets.
Metaheuristic	**HGS (Vidal)**	$100 < N < 10,000$	< 0.1%	Robustness; extremely high-quality solutions; open-source availability (PyVRP).	Complex implementation (Split algorithm); slower than SISR on massive instances.
Metaheuristic	**SISR**	$500 < N < 50,000$	~0.15%	Speed; simplicity of implementation; spatial	Slightly less precise than HGS on tightly

				slack concept scales beautifully.	constrained small instances.
Metaheuristic	**ALNS**	$N < 5,000$	0.5% - 1.0%	Flexibility; best for multi-objective or heterogeneous fleet variants.	Requires tuning of many operator weights; generally dominated by HGS/SISR in pure cost.
NCO (Hybrid)	**NDS**	$100 < N < 1,000$	Competitive	Fast convergence; learns instance-specific structures.	Requires GPU for inference; training time overhead.
LLM-Gen	**VRPAgent**	$N > 1,000$	**-0.30%** (New BKS)	Discovers novel operators that outperform human logic on massive graphs.	"Black box" logic of generated code; dependency on LLM API costs.

6.3 Analysis of Results

- **Small/Medium ($N \le 400$):** HGS is the undisputed king of heuristics, often finding the optimal solution (verified by VRPSolver) in seconds.
- **Large ($N \ge 1000$):** The "human design" barrier appears. HGS and SISR struggle to explore the massive neighborhoods effectively. Here, VRPAgent shines. Its AI-generated operators seem to capture high-level structural dependencies that simple "ruin and

recreate" misses, pushing the envelope of solution quality on industrial-scale problems.[36]

---

7. Software Ecosystem: Tools of the Trade

The gap between academic research and industrial application has been bridged by high-quality software libraries.

7.1 PyVRP: The Open Source Standard

PyVRP is the premier open-source library for VRP in 2025.[15]

- *Architecture:* It is a Python wrapper around a highly optimized C++ implementation of HGS. This architecture provides the best of both worlds: the ease of Python for modeling and data handling, and the raw speed of C++ for the heavy lifting (Split algorithm, Local Search).
- *PCVRP Support:* While natively designed for CVRP/VRPTW, PyVRP's flexible "client" and "vehicle type" definitions allow it to model PCVRP. Users can define different vehicle profiles for different days or model the pattern selection by defining multiple "virtual" clients with mutual exclusivity constraints.[39]
- *Community:* The library is actively maintained, with a growing repository of extensions for "Rich" variants like multi-trip and heterogeneous fleets.[41]

7.2 Hexaly (formerly LocalSolver)

Hexaly represents the commercial state-of-the-art. It is not a MIP solver like Gurobi; it is a "global optimization solver" based on local search and constraint programming.[42]

- *List-Based Modeling:* Hexaly introduces a list variable type. A route is simply a list of visited nodes. This abstraction allows users to write constraints ("visit count must be 2") that look like programming logic rather than linear algebra.
- *Performance:* Hexaly claims dominance on massive industrial instances ($N=30,000$), boasting gaps below 3% where traditional solvers fail to even load the model. It is widely used in waste collection and field service optimization.[44]

7.3 Google OR-Tools

OR-Tools is the "Swiss Army Knife" of routing. It uses a Constraint Programming (CP) approach with a Guided Local Search (GLS) metaheuristic layer.[45]

- *Pros:* Extremely flexible. Can handle almost any side constraint (driver breaks, complex reloading) by adding CP propagators.
- *Cons:* Generally slower and produces slightly lower quality solutions than dedicated solvers like HGS or VRPSolver on standard benchmarks. However, for "messy" real-world PCVRPs, its flexibility often outweighs the raw algorithmic performance gap.

8. Rich Variants and Future Directions

The academic PCVRP is an abstraction. Real-world applications introduce "Rich" constraints that require algorithmic adaptation.

8.1 PVRPTW: Adding Time Windows

When Time Windows are added (PVRPTW), the problem becomes tighter. Paradoxically, this can aid Exact methods (by pruning the search tree) but hinder heuristics (by making feasibility checks expensive).

- *Algorithmic Adaptation:* HGS adapts by treating time window violations as soft constraints in the objective function. The population evolves to minimize this penalty, allowing the search to traverse infeasible regions to find the feasible optimum.[26]

8.2 Workload Equity and Driver Consistency

In multi-period planning, human factors are critical. Drivers prefer consistent territories (Driver Consistency) and fair schedules (Workload Equity).

- *Consistency:* Modeled as a constraint where customer i must be visited by the same vehicle k on all visit days.
- *Equity:* Modeled as minimizing the difference between the longest and shortest routes over the horizon.
- *SOTA Approach:* Recent literature (2022-2024) suggests a two-phase approach. Phase 1 solves the PCVRP for pure cost minimization. Phase 2 fixes the routes and solves a Reassignment Problem (a form of Bin Packing) to swap routes between drivers to achieve equity and consistency without degrading the total travel cost.[47]

8.3 Future Trajectories (2026+)

1. **Generative Heuristics:** The success of VRPAgent implies a future where algorithms are not static code but dynamic agents that "write" the optimization logic on the fly for the specific instance at hand.
2. **Convergence of Learning and Exact Methods:** Research is underway to use Graph Neural Networks (GNNs) to guide the branching decisions of BCP solvers. This "Neural Branch-and-Bound" could potentially double the size of solvable instances to $N=300$.[21]
3. **Dynamic/Stochastic Integration:** The static PCVRP is giving way to Dynamic PCVRP, where demands and travel times are stochastic. DRL-based "Look-ahead" policies will likely become the standard for the operational execution of tactical plans generated by HGS.[33]

In conclusion, the Periodic Capacitated Vehicle Routing Problem has matured into a solvable challenge for industrial scales, driven by the triad of **VRPSolver** (Exact), **HGS/PyVRP** (Metaheuristic), and **VRPAgent** (AI-Driven). The synergy of these approaches ensures that logistics planning in 2025 is more efficient, scalable, and intelligent than ever before.

Works cited

1. Vehicle routing problem - Wikipedia, accessed December 12, 2025, https://en.wikipedia.org/wiki/Vehicle_routing_problem
2. (PDF) CAPACITATED VEHICLE ROUTING PROBLEM - ResearchGate, accessed December 12, 2025, https://www.researchgate.net/publication/343777001_CAPACITATED_VEHICLE_ROUTING_PROBLEM
3. Effect of formulations over a Periodic Capacitated Vehicle Routing Problem with multiple depots, heterogeneous fleet, and hard time-windows - NIH, accessed December 12, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC11449318/
4. The vehicle routing problem: state of the art classification and review - Lirias, accessed December 12, 2025, https://lirias.kuleuven.be/retrieve/276649
5. A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems - SciSpace, accessed December 12, 2025, https://scispace.com/pdf/a-hybrid-genetic-algorithm-for-multidepot-and-periodic-58backiyrn.pdf
6. A Hybrid Genetic Algorithm for Multi- Depot and Periodic Vehicle Routing Problems - Cirrelt, accessed December 12, 2025, https://www.cirrelt.ca/documentstravail/cirrelt-2010-34.pdf
7. A multi-start local search matheuristic for the capacitated arc routing problem with irregular services | Request PDF - ResearchGate, accessed December 12, 2025, https://www.researchgate.net/publication/394514398_A_multi-start_local_search_matheuristic_for_the_capacitated_arc_routing_problem_with_irregular_services
8. The period vehicle routing problem with service choice, accessed December 12, 2025, https://transportation.northwestern.edu/docs/research/core-topics/logistics-and-supply-chain-management/Smilowitz_PeriodRouting.pdf
9. New formulation and valid inequalities for a periodic capacitated ..., accessed December 12, 2025, https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0335389
10. Literature review on the vehicle routing problem in the green transportation context, accessed December 12, 2025, https://www.researchgate.net/publication/290781056_LITERATURE_REVIEW_ON_THE_VEHICLE_ROUTING_PROBLEM_IN_THE_GREEN_TRANSPORTATION_CONTEXT
11. An Exact Algorithm for the Green Vehicle Routing Problem - PubsOnLine, accessed December 12, 2025, https://pubsonline.informs.org/doi/10.1287/trsc.2016.0734
12. Analysis of Mathematical Formulations of Capacitated Vehicle Routing Problem and Methods for their Solution, accessed December 12, 2025,

https://ispranproceedings.elpub.ru/jour/article/download/533/279
13. Exact Branch-Price-and-Cut Algorithms for Vehicle Routing | Request PDF - ResearchGate, accessed December 12, 2025, https://www.researchgate.net/publication/334100211_Exact_Branch-Price-and-Cut_Algorithms_for_Vehicle_Routing
14. Solving Vehicle Routing Problems With Intermediate Stops Using VRPSolver Models, accessed December 12, 2025, https://www.math.u-bordeaux.fr/~rsadykov/papers/Roboredo_etall_Networks22.pdf
15. wouda-et-al-2024-pyvrp-a-high-performance-vrp-solver-package - VU Research Portal, accessed December 12, 2025, https://research.vu.nl/ws/portalfiles/portal/388474973/wouda-et-al-2024-pyvrp-a-high-performance-vrp-solver-package.pdf
16. PyVRP: A High-Performance VRP Solver Package - the University of Groningen research portal, accessed December 12, 2025, https://research.rug.nl/en/publications/pyvrp-a-high-performance-vrp-solver-package/
17. VRPAgent: LLM-Driven Discovery of Heuristic Operators for Vehicle Routing Problems, accessed December 12, 2025, https://openreview.net/forum?id=02mBAZjFzp
18. Neural Deconstruction Search for Vehicle Routing Problems - arXiv, accessed December 12, 2025, https://arxiv.org/html/2501.03715v2
19. Capacitated Vehicle Routing Problem formulation - AIMMS How-To, accessed December 12, 2025, https://how-to.aimms.com/Articles/332/332-Formulation-CVRP.html
20. New Benchmark Instances for the Capacitated Vehicle Routing Problem - Optimization Online, accessed December 12, 2025, https://optimization-online.org/wp-content/uploads/2014/10/4597.pdf
21. Exact Branch-Price-and-Cut Algorithms for Vehicle Routing | Semantic Scholar, accessed December 12, 2025, https://www.semanticscholar.org/paper/Exact-Branch-Price-and-Cut-Algorithms-for-Vehicle-Costa-Contardo/660b34fae29ac57c0c2993d6bd2742876edec217
22. Results on Cordeau et al. (2001) MDVRPTW instances | Download Table - ResearchGate, accessed December 12, 2025, https://www.researchgate.net/figure/Results-on-Cordeau-et-al-2001-MDVRPTW-instances_tbl2_230846314
23. Solving vehicle routing problems with intermediate stops using VRPSolver models | Request PDF - ResearchGate, accessed December 12, 2025, https://www.researchgate.net/publication/366107781_Solving_Vehicle_Routing_Problems_With_Intermediate_Stops_Using_VRPSolver_Models
24. What is the biggest instance of VRP solved to optimality?, accessed December 12, 2025, https://or.stackexchange.com/questions/13351/what-is-the-biggest-instance-of-vrp-solved-to-optimality
25. A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems

| Request PDF - ResearchGate, accessed December 12, 2025, https://www.researchgate.net/publication/259928449_A_Hybrid_Genetic_Algorithm_for_Multidepot_and_Periodic_Vehicle_Routing_Problems
26. Hybrid Genetic Search for the Vehicle Routing Problem with Time Windows: a High-Performance Implementation - Wouter Kool, accessed December 12, 2025, https://wouterkool.github.io/pdf/paper-kool-hgs-vrptw.pdf
27. Slack Induction by String Removals for Vehicle Routing Problems - ResearchGate, accessed December 12, 2025, https://www.researchgate.net/publication/338614875_Slack_Induction_by_String_Removals_for_Vehicle_Routing_Problems
28. Slack Induction by String Removals for Vehicle Routing Problems - IDEAS/RePEc, accessed December 12, 2025, https://ideas.repec.org/a/inm/ortrsc/v54y2020i2p417-433.html
29. SISR Slack Induction by String Removals | PDF | Mathematical Optimization - Scribd, accessed December 12, 2025, https://www.scribd.com/document/797815670/SISR-Slack-induction-by-string-removals
30. A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems - IDEAS/RePEc, accessed December 12, 2025, https://ideas.repec.org/a/eee/ejores/v322y2025i2p357-375.html
31. A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems | Request PDF - ResearchGate, accessed December 12, 2025, https://www.researchgate.net/publication/380857051_A_review_and_ranking_of_operators_in_adaptive_large_neighborhood_search_for_vehicle_routing_problems
32. Reinforcement Learning for Solving the Vehicle Routing Problem, accessed December 12, 2025, http://papers.neurips.cc/paper/8190-reinforcement-learning-for-solving-the-vehicle-routing-problem.pdf
33. Deep reinforcement learning for dynamic vehicle routing with demand and traffic uncertainty, accessed December 12, 2025, https://www.researchgate.net/publication/394329039_Deep_reinforcement_learning_for_dynamic_vehicle_routing_with_demand_and_traffic_uncertainty
34. Deep Reinforcement Learning for Dynamic Capacitated Vehicle Routing Problem, accessed December 12, 2025, https://openreview.net/forum?id=Gs8jWk0F01¬eId=4JLV3CIkml
35. NEURAL DECONSTRUCTION SEARCH FOR VEHICLE ROUTING PROBLEMS - OpenReview, accessed December 12, 2025, https://openreview.net/pdf?id=SrnTGdJKYG
36. VRPAGENT: LLM-DRIVEN DISCOVERY OF HEURIS- TIC OPERATORS FOR VEHICLE ROUTING PROBLEMS - OpenReview, accessed December 12, 2025, https://openreview.net/pdf/0633c4d756ab3fce71d49567912ad35b725f14e1.pdf
37. VRPAgent: LLM-Driven Discovery of Heuristic Operators for Vehicle Routing Problems, accessed December 12, 2025, https://chatpaper.com/paper/197292

38. Benchmarks · VROOM-Project/vroom Wiki - GitHub, accessed December 12, 2025, https://github.com/VROOM-Project/vroom/wiki/Benchmarks
39. PyVRP: A High-Performance VRP Solver Package - ResearchGate, accessed December 12, 2025, https://www.researchgate.net/publication/377789326_PyVRP_A_High-Performance_VRP_Solver_Package
40. PyVRP: a high-performance VRP solver package - arXiv, accessed December 12, 2025, https://arxiv.org/pdf/2403.13795
41. PyVRP/PyVRP: Open-source, state-of-the-art vehicle routing problem solver in an easy-to-use Python package. - GitHub, accessed December 12, 2025, https://github.com/PyVRP/PyVRP
42. Hexaly on very large-scale instances of the Capacitated Vehicle Routing Problem (CVRP), accessed December 12, 2025, https://www.hexaly.com/benchmarks/large-scale-instances-capacitated-vehicle-routing-problem-cvrp
43. Capacitated Vehicle Routing Problem (CVRP) - Hexaly, accessed December 12, 2025, https://www.hexaly.com/templates/capacitated-vehicle-routing-problem-cvrp
44. Vooban Optimizes Waste Collection with Hexaly, Improving Fill Rate from 50% to over 80%, accessed December 12, 2025, https://www.hexaly.com/customers/vooban-optimizes-waste-collection-with-hexaly
45. Comparing VRP Solvers: OR-Tools, OptaPlanner, & SaaS APIs - Singdata Lakehouse, accessed December 12, 2025, https://www.singdata.com/trending/comparing-vrp-solvers-ortools-optaplanner-saas/
46. Mathematical models for the periodic vehicle routing problem with time windows and time spread constraints - AccScience Publishing, accessed December 12, 2025, https://accscience.com/journal/IJOCTA/11/1/10.11121/ijocta.01.2021.00899
47. CIRRELT-2022-22, accessed December 12, 2025, https://www.cirrelt.ca/documentstravail/cirrelt-2022-22.pdf
48. Workload equity in multiperiod vehicle routing problems - PolyPublie, accessed December 12, 2025, https://publications.polymtl.ca/63361/1/2025_Nekooghadirli_Workload_Equity_Multiperiod_Vehicle_Routing.pdf
49. Reinforcement Learning for Solving Stochastic Vehicle Routing Problem with Time Windows - arXiv, accessed December 12, 2025, https://arxiv.org/pdf/2402.09765