Advancing Multi-Objective Neural Combinatorial Optimization: A Comprehensive Analysis of GDPO and Hierarchical Frameworks for Periodic Vehicle Routing

1. Introduction

The discipline of combinatorial optimization (CO) stands at a pivotal juncture, precipitated by the rapid maturation of Deep Reinforcement Learning (DRL) and its encroachment upon domains traditionally dominated by exact operations research (OR) methods. For decades, the solution of NP-hard problems such as the Vehicle Routing Problem (VRP) and its complex variants—specifically the Periodic Capacitated Vehicle Routing Problem (PCVRP)—relied on manually handcrafted heuristics, metaheuristics like Genetic Algorithms (GA) or Simulated Annealing (SA), and rigorous exact solvers like Branch-and-Price. While these methods offer theoretical guarantees or high-quality solutions for smaller instances, they suffer from the "curse of dimensionality" and prohibitive computational costs when scaling to real-world industrial magnitudes.[1]

The emergence of Neural Combinatorial Optimization (NCO) promises a paradigm shift: the ability to "learn to construct" solutions. By training neural networks, typically based on the Transformer architecture or Graph Neural Networks (GNNs), to recognize the underlying topological and demand patterns of routing instances, NCO solvers can infer near-optimal solutions in real-time without the need for iterative retraining.[3] However, the transition from single-objective optimization (minimizing distance) to Multi-Objective Combinatorial Optimization (MOCO) presents a formidable theoretical and practical barrier. Real-world logistics do not merely minimize kilometers; they must simultaneously optimize for conflicting objectives such as driver consistency, workload balance, service time window adherence, and fleet minimization.[5]

In this landscape, the standard reinforcement learning approach of scalarizing rewards—summing weighted objective values into a single feedback signal—has proven insufficient. It leads to a phenomenon known as "reward collapse," where the nuanced trade-offs between conflicting objectives are lost in the aggregation process, causing the agent to over-optimize easier objectives while neglecting complex constraints.[7]

This report provides an exhaustive analysis of a novel solution to this problem: **Group**

reward-Decoupled Normalization Policy Optimization (GDPO). Originally developed for aligning Large Language Models (LLMs) with diverse human preferences, GDPO offers a mechanism for handling heterogeneous reward signals that is uniquely applicable to the multi-objective structures of the PCVRP. We dissect the theoretical mechanics of GDPO, contrast it with established MORL frameworks like Preference-Conditioned Multi-Objective Combinatorial Optimization (PMOCO) and Gradient Projection (PCGrad), and synthesize a comprehensive framework for applying these techniques to the hierarchical challenges of periodic routing.[9]

2. The Theoretical Landscape of Multi-Objective Reinforcement Learning

To appreciate the innovation of GDPO, one must first understand the limitations of the incumbent paradigms in Multi-Objective Reinforcement Learning (MORL). The central challenge in MORL is the vector-valued nature of the reward signal. In a standard Markov Decision Process (MDP), the reward r_t is a scalar. In a Multi-Objective MDP (MOMDP), the reward is a vector $\mathbf{r}_t \in \mathbb{R}^m$, where m is the number of objectives. The goal is not to find a single optimal policy, but to approximate the Pareto Frontier—the set of policies where no objective can be improved without degrading another.[11]

2.1 The Scalarization Trap and Reward Collapse

The most straightforward approach to MOMDPs is scalarization, typically using a weighted sum function $f(\mathbf{r}_t, \mathbf{w}) = \sum_{i=1}^m w_i r_{t,i}$, where $\mathbf{w}$ represents the preference vector of the decision-maker. This reduces the problem to a standard single-objective MDP solvable by algorithms like PPO or REINFORCE.[2]

However, recent research into the fine-tuning of generative models has engaged with a critical failure mode of this approach, termed "reward collapse." This phenomenon is particularly acute in Group Relative Policy Optimization (GRPO), a method where the advantage of an action is calculated relative to a group of sampled actions.[7]

In standard GRPO, the rewards for all objectives are summed before normalization.
Let $r_{total}^{(j)} = \sum_{i=1}^m r_i^{(j)}$ for the j-th rollout in a group of size G.
The advantage is computed as:

$$A^{(j)} = \frac{r_{total}^{(j)} - \mu_{group}}{\sigma_{group}}$$
This creates a fundamental resolution issue. Consider a bi-objective VRP scenario where:

- Objective A (Distance Efficiency) yields a reward of 0 or 10.
- Objective B (Driver Consistency) yields a reward of 0 or 5.

A trajectory achieving only A (Total = 10) and a trajectory achieving two B's (Total = 10,

assuming additivity for example) become indistinguishable to the policy gradient. The distinction between "highly efficient but inconsistent" and "highly consistent but inefficient" is obliterated. Furthermore, if the variance of Objective A is significantly higher than Objective B, the normalization parameters ($\mu_{group}, \sigma_{group}$) will be dominated by A. The signal from B becomes statistical noise, effectively vanishing from the gradient. The agent learns to ignore the lower-variance objective entirely, a pathology observed in both LLM alignment and logistics optimization.[3]

2.2 Neural Combinatorial Optimization and the Pareto Front

In the specific context of NCO, scalarization poses an additional geometric problem. If the Pareto frontier of the VRP is non-convex (which is common in discrete combinatorial spaces), a static linear weighted sum cannot theoretically discover solutions in the "concave" regions of the front. This limitation has driven the development of decomposition-based methods (MOEA/D-DRL) and Tchebycheff scalarization functions, which can target these non-convex regions.[13]

However, even with Tchebycheff scalarization, the training stability remains tethered to the reward magnitudes. If an NCO model is trained to minimize total tour length (values $\sim 10^3$) and maximize load balance (Gini coefficient $\in $), the gradients derived from the tour length error will be orders of magnitude larger than those from the balance error. Without meticulous, manual reward shaping and clipping—a brittle and non-generalizable process—the neural network will fail to converge on a balanced policy.[15]

3. GDPO: Group Reward-Decoupled Normalization Policy Optimization

GDPO emerges as a structural remedy to the limitations of scalarization and aggregation-based normalization. The core insight of GDPO is that normalization must occur *independently* for each objective channel to preserve the relative signal strength of each goal, regardless of its absolute magnitude or variance.[7]

3.1 Mechanism of Decoupled Normalization

GDPO fundamentally alters the computation of the advantage function. Instead of aggregating rewards and then normalizing, GDPO performs group-wise normalization on each reward component separately.

For a set of G rollouts generated by the current policy π_θ in response to a state s, and for each objective $k \in \{1, \dots, m\}$:

1. Component Normalization: Calculate the mean μ_k and standard deviation σ_k of the k-th reward across the G rollouts.

$$\hat{A}_k^{(j)} = \frac{r_k^{(j)} - \mu_k}{\sigma_k + \epsilon}$$
Here, $\hat{A}_k^{(j)}$ represents the normalized advantage of the j-th rollout specifically regarding objective k. This effectively converts the raw reward into a Z-score relative to the local batch. Whether the raw reward is in the range$$ or $$, the normalized advantage is scaled to a standard normal distribution.

2. Aggregation: The final advantage used for the policy update is the sum of these normalized components.

   $$A_{GDPO}^{(j)} = \sum_{k=1}^m \hat{A}_k^{(j)}$$
3. **Batch-wise Stability**: To ensure that the magnitude of the gradient does not grow linearly with the number of objectives (which would destabilize the learning rate), the aggregated advantages A_{GDPO} are often re-normalized across the batch.[8]

3.2 Advantages Over Traditional Methods

The implications of this reordering are profound for multi-objective optimization:

- **Signal Preservation**: GDPO preserves the "distinct advantage groups." In the binary example discussed in Section 2.1, GDPO would assign different vector coordinates to the (A only) and (B only) cases before summation, preventing them from collapsing into a single scalar value if their distributions differ. Empirical evidence from LLM tasks shows that GDPO maintains a significantly higher number of distinct advantage values compared to GRPO, providing a higher-resolution training signal.[8]
- **Scale Invariance**: By normalizing each objective to a Z-score, GDPO automatically handles the disparate scales of VRP objectives (e.g., distance in kilometers vs. consistency in percentage). This eliminates the need for manual weight tuning to balance gradient magnitudes, acting as an adaptive, dynamic scaling mechanism.[9]
- **Priority Handling via Conditioning**: GDPO facilitates a nuanced handling of objective priorities through "conditioned rewards." Rather than simple weighting, rewards for secondary objectives (e.g., efficiency) can be structurally conditioned on the satisfaction of primary objectives (e.g., feasibility). This prevents the "easier reward first" pathology where agents maximize simple but trivial rewards while ignoring difficult constraints.[7]

4. The Periodic Capacitated Vehicle Routing Problem (PCVRP)

To ground the application of GDPO, we must rigorously define the target domain. The Periodic Capacitated Vehicle Routing Problem (PCVRP) is a generalization of the classical VRP that extends the planning horizon over a period T (e.g., a week or a month). It introduces a layer of tactical planning (scheduling) atop the operational planning (routing).[19]

4.1 Problem Formulation and Complexity

The PCVRP is defined on a complete graph $G = (V, E)$. The node set $V = \{0, 1, \dots, N\}$ consists of a depot (0) and N customers.

- **Periodicity**: The planning horizon consists of T days.
- **Service Frequency**: Each customer i has a service frequency f_i (number of visits required over T) and a set of allowable visit patterns P_i. For example, if $T=5$ and $f_i=2$, a pattern might be $\{Monday, Thursday\}$.
- **Capacity**: A fleet of homogeneous or heterogeneous vehicles with capacity Q is available each day.
- **Demand**: Customer i has demand q_i which must be serviced on the visited days.

The optimization problem is hierarchical and decomposed into three sub-problems [21]:

1. **Pattern Assignment**: Selecting a valid pattern $p \in P_i$ for each customer.
2. **Day Assignment**: Implicitly determined by the pattern.
3. **Vehicle Routing**: Solving a Capacitated VRP (CVRP) for each day $t \in \{1, \dots, T\}$ for the customers assigned to that day.

The complexity of PCVRP is significantly higher than CVRP. It is not merely a sequence of independent VRPs; the pattern assignment couples the days together. A decision to visit a customer on Day 1 constrains the system to visit them again on Day 4 (if the pattern is 1-4), potentially creating an infeasible or highly inefficient routing scenario on Day 4. This long-term dependency makes it a prime candidate for Reinforcement Learning, which is designed to optimize sequential decision processes.[23]

4.2 Multi-Objective Nature of PCVRP

While academic benchmarks often focus on minimizing total distance, real-world PCVRP applications in waste management, vending machine replenishment, and home healthcare are inherently multi-objective.[6]

1. **Operational Cost**: Minimizing total travel distance and vehicle usage.
2. **Driver Consistency**: In service industries, it is crucial that the same driver visits the same customer across the period to build trust and service knowledge. This requires the routing algorithm to maintain consistent assignments, often at the expense of optimal routing geometry.[23]
3. **Workload Balance**: Ensuring that the workload (time or distance) is evenly distributed across days and drivers. Extreme peaks in workload lead to overtime costs and operational risk.[26]
4. **Visual Compactness**: Human planners prefer routes that are visually distinct and non-overlapping, even if they are slightly suboptimal in distance.

These objectives are heterogeneous in nature (continuous cost, discrete consistency counts, statistical balance metrics) and conflicting, creating precisely the "reward collapse"

environment GDPO is designed to resolve.

5. Integrating GDPO into Neural Combinatorial Optimization

The synthesis of GDPO with NCO architectures offers a robust pathway to solving the multi-objective PCVRP. We propose a framework that integrates GDPO into a hierarchical Deep Reinforcement Learning model.

5.1 Hierarchical Architecture for PCVRP

Given the two-stage nature of PCVRP (Scheduling + Routing), a hierarchical RL (HRL) approach is superior to a monolithic model.

- **Upper-Level Agent (Scheduler)**: Responsible for Pattern Assignment. It receives the global state (all customer locations, demands, frequencies) and outputs a pattern assignment for each customer.
- **Lower-Level Agent (Router)**: Responsible for solving the daily CVRPs. It receives the subset of customers assigned to a specific day and outputs the routing sequence.[27]

In this framework, the Lower-Level Agent can be a pre-trained constructive heuristic (like the Attention Model or POMO) that is relatively static. The primary learning challenge lies with the Upper-Level Agent, which must learn to assign patterns that minimize the *aggregate* cost of the daily routes while maximizing consistency and balance.[27]

5.2 Applying GDPO to the Scheduler

The Upper-Level Agent's training is where GDPO becomes critical. The reward signal for a pattern assignment is a vector $\mathbf{R} =$.

- R_{cost}: The sum of routing costs for all days (derived from the Lower-Level Agent).
- $R_{consistency}$: A penalty metric for driver switches.
- $R_{balance}$: A metric (e.g., negative variance) of daily fleet usage.

If we use standard GRPO or PPO with weighted sums, the R_{cost} (which might be in the thousands) will dwarf $R_{consistency}$ (which is a count) and $R_{balance}$ (which is a ratio). Even with static scaling, the *variance* of the routing cost (which depends on the complex geometry of the daily VRPs) will introduce significant noise.

GDPO Implementation steps:

1. **Group Sampling**: The Scheduler generates G different pattern assignments for the same problem instance (using nucleus sampling or diverse beam search).
2. **Parallel Evaluation**: The Lower-Level Agent solves the daily VRPs for all G assignments to generate the raw reward vectors.

3. **Decoupled Normalization**:
    - Normalize the R_{cost} values *across the group*. This identifies which assignment was relatively most efficient, filtering out the baseline difficulty of the instance.
    - Normalize $R_{consistency}$ values *across the group*.
    - Normalize $R_{balance}$ values *across the group*.
4. **Update**: The Scheduler's policy is updated using the sum of these normalized advantages.

This ensures that a small improvement in Consistency (which might be critical for the user) is not ignored simply because the routing cost has high variance. It allows the agent to learn the subtle structural correlations between pattern choices and consistency outcomes.[7]

5.3 Handling Constraints via Conditional Rewards

The GDPO literature suggests "conditional rewards" for priority management. In PCVRP, feasibility (capacity constraints) is paramount.
We can define a conditional reward structure:

$$R_{total} = R_{feasibility} + \mathbb{I}(R_{feasibility}=1) \cdot (w_1 \hat{A}_{cost} + w_2 \hat{A}_{consistency})$$

Here, the agent only receives the multi-objective optimization signal if the basic feasibility constraint is met. GDPO's normalization is applied to the components within the valid group, ensuring that among feasible solutions, the trade-offs are optimized efficiently.7

6. Alternative Algorithms and Comparative Analysis

While GDPO addresses the reward signal processing, other advanced algorithms tackle different aspects of the multi-objective learning problem. A comprehensive research report must contextualize GDPO against these alternatives.

6.1 Preference-Conditioned MOCO (PMOCO)

PMOCO represents the state-of-the-art in "Pareto Set Learning." Instead of training a single policy for a fixed trade-off, PMOCO trains a "Hypernetwork" or a "Conditioned Decoder" that takes a preference vector $\mathbf{w}$ as input and generates the policy parameters $\theta_\mathbf{w}$ dynamically.[15]

- **Mechanism**: The architecture typically involves a shared encoder for the problem instance and a lightweight hypernetwork that modulates the attention weights of the decoder based on $\mathbf{w}$.
- **Strengths**: A single model can generate the entire Pareto front at inference time by sweeping $\mathbf{w}$.
- **Weakness relative to GDPO**: PMOCO usually relies on weighted sum scalarization

during training ($R = \mathbf{w} \cdot \mathbf{r}$). It remains vulnerable to the scale imbalance and reward collapse issues.
- **Synergy**: GDPO and PMOCO are orthogonal. One can use the PMOCO *architecture* (preference conditioning) combined with the GDPO *training algorithm* (decoupled normalization). In this hybrid, the normalized advantages would be weighted by $\mathbf{w}$ before aggregation, ensuring that the preference vector steers the policy through a normalized, stable reward landscape.[2]

6.2 Gradient Projection (PCGrad)

PCGrad (Projected Conflicting Gradients) operates at the optimization level, modifying the gradients themselves rather than the rewards.[30]

- **Mechanism**: If the gradients of two tasks (or objectives) g_1 and g_2 conflict (i.e., their cosine similarity is negative), PCGrad projects g_1 onto the normal plane of g_2, removing the conflicting component.
- **Application to PCVRP**: This is highly effective when objectives are strictly competitive (e.g., minimizing distance vs. maximizing load balance often pull parameters in opposite directions). PCGrad prevents "catastrophic interference" where learning one task unlearns the other.
- **Computational Cost**: PCGrad requires computing and storing individual gradients for each objective for every parameter, which can be prohibitively expensive for large Transformer models (VRP models often have millions of parameters) compared to the scalar manipulations of GDPO.[32]

6.3 GradNorm

GradNorm addresses the imbalance in learning speeds between different objectives.

- **Mechanism**: It dynamically adjusts the weights of the loss functions for different objectives to ensuring that the gradients for each objective have similar magnitudes.[33]
- **Comparison**: GradNorm is similar in spirit to GDPO (balancing signals) but operates on the loss/gradient magnitude, whereas GDPO operates on the reward distribution. GDPO is generally more stable for RL because it handles the *variance* of the rewards (Z-scoring), whereas GradNorm focuses on the gradient norm which can be noisy in RL.[4]

6.4 Evolutionary-RL Hybrids (AGE-MORL)

AGE-MORL combines RL with Multi-Objective Evolutionary Algorithms (MOEA).

- **Mechanism**: An RL agent acts as a "hyper-heuristic," selecting which evolutionary operators (crossover, mutation) to apply to the current population to maximize the spread and quality of the Pareto front.[5]
- **PCVRP Fit**: Since PCVRP has a discrete and disjoint solution space (due to pattern assignments), evolutionary methods are traditionally strong. Using RL to guide the search

(Learning to Search) rather than construct the solution (Learning to Construct) is a viable alternative, though typically slower at inference time than end-to-end NCO.[35]

7. The Frontier: Foundation Models and Zero-Shot Generalization

The future of applying algorithms like GDPO to logistics lies in **Generalist Vehicle Routing Models**. Recent works like **RouteFinder** propose training a single "Foundation Model" on a massive variety of VRP variants (CVRP, VRPTW, PDVRP, etc.) simultaneously.[37]

In this context, different VRP variants can be viewed as different "objectives" or "tasks" in a Multi-Task Learning (MTL) framework.

- **Zero-Shot Generalization**: A model trained with GDPO on a mix of distance and time-window rewards can potentially generalize to a new, unseen constraint profile by treating it as a new preference vector configuration.[4]
- **Unified Attributes**: By encoding problem constraints (time windows, periodicity) as "attribute embeddings" (similar to prompts in LLMs), the model learns the *concept* of a constraint. GDPO ensures that the model does not overfit to the most common attributes (like capacity) while failing to learn rarer ones (like periodicity).[33]

8. Case Study Simulation: Optimizing a Waste Collection Network

To illustrate the practical utility of these concepts, consider a PCVRP scenario for municipal waste collection.

- **Scenario**: 1000 collection points, 5-day horizon.
- **Objectives**: Minimize Cost (fuel/wages), Maximize Consistency (residents prefer same driver), Minimize Missed Pickups (reliability).
- **Challenge**: "Missed Pickups" is a sparse, binary failure signal. "Cost" is a dense, continuous signal.
- **Failure of Standard RL**: The agent quickly learns to minimize cost. It occasionally misses a pickup, but the penalty is drowned out by the massive reward from fuel savings. The policy converges to a low-cost, low-reliability state.
- **GDPO Solution**: The reward for "Missed Pickups" is normalized independently. A trajectory with 0 misses gets a high positive advantage relative to one with 1 miss, *even if* the fuel cost is higher. The decoupled normalization forces the agent to distinguish the "Reliability" axis explicitly. The "Conditioned Reward" mechanism further enforces that fuel optimization only counts *if* missed pickups are zero.

9. Conclusion

The integration of **GDPO** into the domain of combinatorial optimization represents a significant maturation of Neural Combinatorial Optimization. By addressing the fundamental statistical flaws in how multi-objective rewards are aggregated, GDPO provides the stability required to train agents on the highly complex, heterogeneous objectives of the **Periodic Capacitated Vehicle Routing Problem**.

While specialized architectures like **PMOCO** provide the structural capacity for Pareto exploration, and **PCGrad** offers protection against gradient interference, GDPO offers a computationally efficient, drop-in enhancement for the reward processing layer that is critical for real-world deployment. The convergence of these methods points toward a future of "Foundation Routing Models"—systems capable of zero-shot generalization across the messy, multi-faceted constraints of global logistics, driven by the same alignment techniques that have powered the revolution in generative AI.

The transition from "solving a VRP instance" to "aligning a VRP policy with human preferences" is the next great frontier in Operations Research, and GDPO is a primary engine of this evolution.

Table 1: Comparative Analysis of MORL Algorithms for PCVRP

Algorithm	Primary Mechanism	Best Application in PCVRP	Computational Overhead	Key Reference
GDPO	Decoupled Reward Normalization	Handling heterogeneous rewards (e.g., Consistency vs. Cost)	Low (Scalar operations)	7
PMOCO	Preference-Conditioned Hypernetworks	Generating continuous Pareto fronts for trade-off analysis	Medium (Hypernetwork inference)	2
PCGrad	Gradient Orthogonalization	Scenarios with strictly competing objectives (interference)	High (Per-parameter gradient storage)	30

GradNorm	Loss Weight Normalization	Balancing learning rates across diverse constraints	Medium (Gradient norm computation)	[4]
AGE-MORL	RL-Guided Evolutionary Search	Discrete Pattern Assignment (high complexity search)	High (Population evaluation)	[5]

Table 2: PCVRP Problem Structure and RL Components

PCVRP Component	Decision Type	RL Agent Role	Objective Nature
Pattern Assignment	Discrete Selection (P_i)	Upper-Level Policy (Scheduler)	Strategic / Long-term
Clustering	Grouping (V_t)	Implicit / Joint with Pattern	Balanced / Distributional
Routing	Sequence Permutation (π)	Lower-Level Policy (Router)	Operational / Geometric
Evaluation	Horizon Aggregation	Critic / Value Function	Multi-objective / Noisy

Works cited

1. Learning heuristic selection using a Time Delay Neural Network for Open Vehicle Routing | Request PDF - ResearchGate, accessed January 21, 2026, https://www.researchgate.net/publication/316095940_Learning_heuristic_selection_using_a_Time_Delay_Neural_Network_for_Open_Vehicle_Routing
2. Collaborative deep reinforcement learning for solving multi-objective vehicle routing problems - Institutional Knowledge (InK) @ SMU, accessed January 21, 2026, https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=10328&context=sis_research
3. GDPO: Group reward-Decoupled Normalization Policy Optimization for

Multi-reward RL Optimization | alphaXiv, accessed January 21, 2026, https://www.alphaxiv.org/overview/2601.05242
4. Multi-Task Learning for Routing Problem with Cross-Problem Zero-Shot Generalization | Request PDF - ResearchGate, accessed January 21, 2026, https://www.researchgate.net/publication/383466617_Multi-Task_Learning_for_Routing_Problem_with_Cross-Problem_Zero-Shot_Generalization
5. AGE-MORL: Agent-Guided Evolutionary Control for Multi-Objective Reinforcement Learning, accessed January 21, 2026, https://openreview.net/forum?id=lxPe6S5E7k
6. Generation of distribution routes with shorter distances and fewer vehicles using the simulated annealing algorithm - ResearchGate, accessed January 21, 2026, https://www.researchgate.net/publication/397429376_Generation_of_distribution_routes_with_shorter_distances_and_fewer_vehicles_using_the_simulated_annealing_algorithm
7. [Literature Review] GDPO: Group reward-Decoupled Normalization Policy Optimization for Multi-reward RL Optimization - Moonlight | AI Colleague for Research Papers, accessed January 21, 2026, https://www.themoonlight.io/review/gdpo-group-reward-decoupled-normalization-policy-optimization-for-multi-reward-rl-optimization
8. GDPO: Group reward-Decoupled Normalization Policy Optimization ..., accessed January 21, 2026, https://nvlabs.github.io/GDPO/
9. GDPO: Group reward-Decoupled Normalization Policy Optimization for Multi-reward RL Optimization - arXiv, accessed January 21, 2026, https://arxiv.org/html/2601.05242v1
10. RETHINKING NEURAL MULTI-OBJECTIVE COMBINATO- RIAL OPTIMIZATION VIA NEAT WEIGHT EMBEDDING - ICLR Proceedings, accessed January 21, 2026, https://proceedings.iclr.cc/paper_files/paper/2025/file/78efbc5386c5a7c241e7fcc482d3c3dc-Paper-Conference.pdf
11. Pareto-Optimal Multi-Objective RL - Emergent Mind, accessed January 21, 2026, https://www.emergentmind.com/topics/pareto-optimal-multi-objective-reinforcement-learning
12. A Deep Reinforcement Learning Model to Solve the Stochastic Capacitated Vehicle Routing Problem with Service Times and Deadlines - MDPI, accessed January 21, 2026, https://www.mdpi.com/2227-7390/13/18/3050
13. Deep Reinforcement Learning for Solving Multi-objective Vehicle Routing Problem, accessed January 21, 2026, https://www.researchgate.net/publication/372755300_Deep_Reinforcement_Learning_for_Solving_Multi-objective_Vehicle_Routing_Problem
14. A Neural Multi-Objective Capacitated Vehicle Routing Optimization Algorithm Based on Preference Adjustment - MDPI, accessed January 21, 2026, https://www.mdpi.com/2079-9292/12/19/4167
15. Pareto Set Learning for Neural Multi-Objective Combinatorial Optimization | OpenReview, accessed January 21, 2026, https://openreview.net/forum?id=QuObT9BTWo
16. Conditional Neural Heuristic for Multi-objective Vehicle Routing Problems -

MARMot Lab, accessed January 21, 2026, https://marmotlab.org/publications/59-TNNLS2024-MOVRP.pdf
17. GDPO: Group reward-Decoupled Normalization Policy Optimization for Multi-reward RL Optimization - ResearchGate, accessed January 21, 2026, https://www.researchgate.net/publication/399596515_GDPO_Group_reward-Decoupled_Normalization_Policy_Optimization_for_Multi-reward_RL_Optimization
18. Paper page - GDPO: Group reward-Decoupled Normalization Policy Optimization for Multi-reward RL Optimization - Hugging Face, accessed January 21, 2026, https://huggingface.co/papers/2601.05242
19. Effect of formulations over a Periodic Capacitated Vehicle Routing Problem with multiple depots, heterogeneous fleet, and hard time-windows - NIH, accessed January 21, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC11449318/
20. A Bibliometric Visualized Analysis and Classification of Vehicle Routing Problem Research, accessed January 21, 2026, https://www.mdpi.com/2071-1050/15/9/7394
21. Modeling and solving an integrated periodic vehicle routing and capacitated facility location problem in the context of solid waste collection - ResearchGate, accessed January 21, 2026, https://www.researchgate.net/publication/391852659_Modeling_and_solving_an_integrated_periodic_vehicle_routing_and_capacitated_facility_location_problem_in_the_context_of_solid_waste_collection
22. Fleet-sizing for multi-depot and periodic vehicle routing problems using a modular heuristic algorithm | Request PDF - ResearchGate, accessed January 21, 2026, https://www.researchgate.net/publication/264980284_Fleet-sizing_for_multi-depot_and_periodic_vehicle_routing_problems_using_a_modular_heuristic_algorithm
23. Research | Manuel Laguna | University of Colorado Boulder, accessed January 21, 2026, https://www.colorado.edu/faculty/laguna/research
24. Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model | Request PDF - ResearchGate, accessed January 21, 2026, https://www.researchgate.net/publication/220289551_Solving_the_vehicle_routing_problem_with_time_windows_and_multiple_routes_exactly_using_a_pseudo-polynomial_model
25. A Multi-Stage Algorithm for a Capacitated Vehicle Routing Problem with Time Constraints, accessed January 21, 2026, https://www.mdpi.com/1999-4893/11/5/69
26. Genetic Algorithm Optimization of Sales Routes with Time and Workload Objectives - MDPI, accessed January 21, 2026, https://www.mdpi.com/2673-9909/5/3/103
27. Hierarchical Learning-based Graph Partition for Large-scale Vehicle Routing Problems | Request PDF - ResearchGate, accessed January 21, 2026, https://www.researchgate.net/publication/388954586_Hierarchical_Learning-based_Graph_Partition_for_Large-scale_Vehicle_Routing_Problems
28. A Simulation-based Optimization Approach for Integrated Outpatient Flow and

Medication Management - Scholarship@Miami, accessed January 21, 2026, https://scholarship.miami.edu/view/pdfCoverPage?instCode=01UOML_INST&filePid=13386208450002976&download=true
29. Preference-Driven Multi-Objective Combinatorial Optimization with Conditional Computation, accessed January 21, 2026, https://arxiv.org/html/2506.08898v2
30. Safe Reinforcement Learning to Make Decisions in Robotics - People @EECS, accessed January 21, 2026, https://people.eecs.berkeley.edu/~shangding.gu/papers/PhD_Dissertation_Shangding_Gu_2024.pdf
31. (PDF) Fantastic Multi-Task Gradient Updates and How to Find Them In a Cone, accessed January 21, 2026, https://www.researchgate.net/publication/388658100_Fantastic_Multi-Task_Gradient_Updates_and_How_to_Find_Them_In_a_Cone
32. Task Weighting through Gradient Projection for Multitask Learning - arXiv, accessed January 21, 2026, https://arxiv.org/html/2409.01793v1
33. Multi-Task Learning for Routing Problem with Cross-Problem Zero-Shot Generalization - arXiv, accessed January 21, 2026, https://arxiv.org/pdf/2402.16891
34. A Multi-Task Dynamic Weight Optimization Framework Based on Deep Reinforcement Learning - MDPI, accessed January 21, 2026, https://www.mdpi.com/2076-3417/15/5/2473
35. Learning-Aided Neighborhood Search for Vehicle Routing Problems - IEEE Computer Society, accessed January 21, 2026, https://www.computer.org/csdl/journal/tp/2025/07/10938384/25mYB3DLUpW
36. A Reinforcement Learning Hyper-Heuristic with Cumulative Rewards for Dual-Peak Time-Varying Network Optimization in Heterogeneous Multi-Trip Vehicle Routing - MDPI, accessed January 21, 2026, https://www.mdpi.com/1999-4893/18/9/536
37. RouteFinder: Towards Foundation Models for Vehicle Routing Problems - EUR Research Information Portal, accessed January 21, 2026, https://pure.eur.nl/ws/portalfiles/portal/208853467/5124_RouteFinder_Towards_Found.pdf
38. A Foundation Model for Vehicle Routing Problems 1 INTRODUCTION - TRISTAN 2025, accessed January 21, 2026, https://tristan2025.org/proceedings/TRISTAN2025_ExtendedAbstract_247.pdf
39. RouteFinder: Towards Foundation Models for Vehicle Routing Problems - arXiv, accessed January 21, 2026, https://arxiv.org/html/2406.15007v4