Adaptive Neuro-Evolution for Dynamic Multi-UAV Routing with Load-Dependent Constraints

Xiaoyan Gong[1,2], Xiaoying Yang[1,2], Zhenwei Wang[1,2], Fuhua Jia[3], Siyuan He [1,2], Tianxiang Cui[1,2], Ruibin Bai[1,2,*]
[1] School of Computer Science, University of Nottingham Ningbo China
[2] The Digital Port Technologies Lab, University of Nottingham Ningbo China
[3] Control System Lab, University of Nottingham Ningbo China

***Abstract*—Energy Capacitated Vehicle Routing Problems with Time Windows (ECVRPTW) are central to modern logistics, particularly UAV-based delivery, owing to the generality of their operational constraints. Existing methods predominantly assume static customer demands, yet real-world operations are inherently dynamic: new service requests emerge during route execution. Conventional static solvers accommodate such changes via customer insertion into ongoing routes—an approach that sacrifices flexibility and often yields suboptimal solutions. Although reinforcement learning (RL) offers rapid inference for dynamic settings, standard models struggle to scale or satisfy complex non-linear constraints. To bridge this gap, we propose a closed-loop neuro-evolutionary framework using a hierarchical manager–worker architecture. The manager (a deep RL agent) determines in real time when to trigger re-optimization. The worker then executes a dual-path process: a Transformer model constructs global route topologies, which are then refined by an evolutionary module to ensure all complex constraints are met. Experimental results demonstrate that our framework effectively adapts to uncertain demands, consistently outperforming static replanning strategies and setting a new state-of-the-art for online routing.**

***Index Terms*—Dynamic VRP, Multi-UAV Task Allocation, Neuro-Evolutionary Algorithm, Reinforcement Learning**

I. Introduction

Unmanned Aerial Vehicles (UAVs) have emerged as a key enabler of next-generation logistics systems, particularly in applications requiring rapid response, such as emergency supply distribution [1], last-mile delivery [2], [3], and on-demand urban logistics [4]. These scenarios naturally lead to vehicle routing problems characterized by multiple operational constraints, including limited onboard energy [5], payload capacity [6], and customer time windows [7]. Real-world UAV logistics systems operate in highly dynamic environments with on-the-fly customer requests [8]. In this context, the routing problem transforms into a continuous management of physical states. As a UAV executes a mission, its operational envelope shrinks irreversibly: battery power is consumed not just by distance, but by the instantaneous gross weight (comprising the UAV's tare weight and payload) and flight maneuvering [9]. Consequently, flight commitments are temporally irreversible; a decision made at an earlier timestamp physically constrains the reachable solution space in the future. When coupled with the stochastic nature of real-world demand—where requests emerge dynamically as shown in Figure 1—the system faces a rigid physical reality that cannot be simply reset or ignored. This imposes a fundamental trilemma between optimality, real-time responsiveness, and physical feasibility.

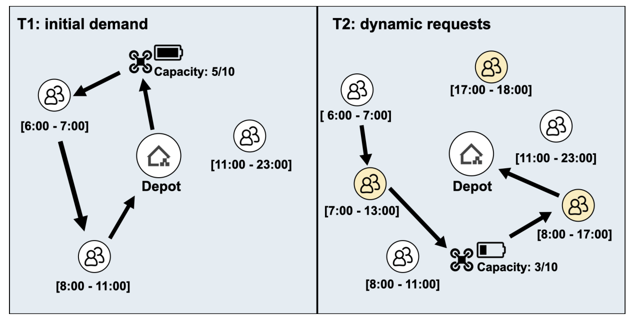

Fig. 1. Schematic representation of the D-ECVRPTW problem. Time T_1 shows the initial mission state with static demands. Time T_2 illustrates the dynamic arrival of new requests (yellow) and the simultaneous evolution of UAV states (reduced battery and updated capacity), requiring an adaptive re-optimization of the remaining routes.

Existing approaches to this challenge have primarily utilized Operations Research (OR), Evolutionary Computation (EC), or Deep Reinforcement Learning (DRL) methods. However, as summarized in Table I, a distinct gap exists in addressing the intersection of dynamic capability and rigorous physical constraints. Traditional exact methods, such as Integer Programming (IP) and Dynamic Programming (DP), search on the exact physical manifold to guarantee global optimality in static, small-scale instances [10], [11]. Yet, when extended to dynamic environments, the combinatorial explosion renders real-time re-optimization computationally infeasible. To address scalability, meta-heuristic approaches have been widely adopted. Studies such as TERRAN [12], 2-stage-ALNS [13], ILNS [14], and EALNS [15] successfully handle complex constraints like capacity and energy but are limited to static environments. Similarly, multi-objective and hybrid evolutionary frameworks, including MOEA [16], Route-then-

*Corresponding author
This work is supported in part by Ningbo Science and Technology Bureau (Project ID 2025QL055, 2025Z197, 2023Z237 and 2025S114), in part by the Basic and Commonweal Programme of Zhejiang Natural Science Foundation (Project ID LY24F020006), in part by UNNC Control System Lab, and in part by Ningbo Digital Port Technologies Key Lab.

TABLE I
COMPARISON OF RELATED UAV ROUTING STUDIES

Problem	Method	Env.	Cap.	Egy.	Load.	TW
EVRP-TW	TERRAN [12]	Sta.	✓	✓	-	✓
EVRPTW-BSU	2-stage-ALNS [13]	Sta.	✓	✓	-	✓
mPDPD-mT	ILNS [14]	Sta.	✓	✓	-	✓
AF-MUTALC	MOEA [16]	Sta.	✓	✓	✓	-
CEVRP-PTW-TR	EALNS [15]	Sta.	✓	✓	-	✓
CEVRP	DRLHH [30]	Sta.	✓	✓	-	-
EVRPTWs	RL [24]	Sta.	-	✓	-	✓
EVRPTW-TOU	SOC [19]	Sta.	✓	✓	✓	✓
ECVRP	ACO [34]	Sta.	✓	✓	✓	-
ECVRP	BHGA [35]	Sta.	✓	✓	✓	-
DVRPSD	RL-Ptr [20]	Dyn.	-	-	-	-
MD-DVRPTW	ALNS [36]	Dyn.	✓	-	-	✓
DVRPTW	GPHH [37]	Dyn.	✓	-	-	✓
DVRPTW	ACS-KM [28]	Dyn.	✓	-	-	✓
DVRPTW	AGA-ES-TW [9]	Dyn.	✓	-	-	✓
DDTWVRP	MSA [32]	Dyn.	✓	-	-	✓
DVRPTW	2-phase-MH [33]	Dyn.	-	-	-	✓
DS-CVRPTW	sMMDP [25]	Dyn.	✓	-	-	✓
***D*-ECVRPTW**	**Proposed**	**Dyn.**	✓	✓	✓	✓

Note: **Env.**: Environment (Static vs. Dynamic); **Cap.**: Vehicle Capacity; **Egy.**: Energy Constraint; **Load.**: Load-dependent Dynamics; **TW**: Time Window.

Schedule [17], memetic programming [18], and SOC [19], incorporate load-dependent dynamics or time windows, yet they lack the responsiveness required for on-the-fly task insertion. Conversely, to enable real-time decision-making, researchers have increasingly turned to Learning-Based Methods. Approaches such as RL-Ptr [20], GNN-PPO [21], [22], GAS [23], and RL [24]–[27] offer millisecond-level inference speeds suitable for dynamic routing. However, these end-to-end models operate primarily as pattern matchers and often lack the deductive reasoning capability to strictly satisfy hard physical constraints, such as integral-based energy consumption. While recent hybrid works like ACS-KM [28], DRL-HH [29]–[31] ,MSA [32], and dynamic genetic algorithms [9], [33] attempt to bridge this gap, they often simplify the physical model—ignoring load-dependent energy dynamics—to maintain computational viability. Consequently, there remains a lack of unified frameworks capable of handling high-dynamic task injection while rigorously adhering to non-linear physical energy constraints and service time windows.

To bridge this gap, we propose the Hybrid Neuro-Evolutionary Search (HNES) framework. Formulated as a neuro-symbolic cognitive architecture, HNES synthesizes the rapid inference of neural networks with the rigorous constraint handling of evolutionary computation. By synergizing a generative topological prior for global search with a physics-informed constraint reasoner for local feasibility, HNES accurately solves non-linear energy constraints that purely neural methods often overlook. Our contributions are as follows:

1) We propose a neuro-symbolic cognitive architecture that mimics dual-process cognition. A High-Level RL Manager learns to arbitrate the timing of optimization, dynamically switching between a Generative Topological Prior for rapid structural adaptation and a Physics-Informed Constraint Reasoner for precise feasibility repair.

2) We introduce the Tripartite Chromosome Encoding to strictly model the irreversibility of dynamic routing. This mechanism segregates the solution into Immutable History, Locked Commitment, and Future Queue, ensuring that all algorithmic modifications strictly adhere to causal consistency and load-dependent energy dynamics.

3) We conduct extensive experiments on dynamic Solomon and Homberger instances, demonstrating that HNES achieves a Pareto-optimal balance. It significantly outperforms pure Neural Combinatorial Optimization baselines in constraint satisfaction and surpasses traditional dynamic heuristics in solution quality and convergence speed.

II. PROBLEM FORMULATION

We study the Dynamic Energy Capacitated Vehicle Routing Problem with Time Windows (D-ECVRPTW) on a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. The node set $\mathcal{N} = \{0\} \cup \mathcal{V}$ comprises the depot (node 0) and the set of all potential customers $\mathcal{V}$. The arc set $\mathcal{A}$ represents feasible flight connections. To capture operational realism, the model rests on four key assumptions: 1) the environment is dynamic, where customer requests are partitioned into an active set $\mathcal{V}^a$ (known) and a pending set $\mathcal{V}^p$ (unknown); 2) flight execution is non-preemptive and irreversible; 3) flight dynamics are load-dependent, meaning energy consumption is coupled with the instantaneous payload; and 4) UAVs operate at a constant cruise speed.

TABLE II
MATHEMATICAL NOTATIONS

Symbol	Description
$\mathcal{N}$	Set of all nodes (depot $\{0\} \cup \mathcal{V}$)
$\mathcal{V}^{\text{a}}$	Set of currently **active** customers (known requests)
$\mathcal{K}$	Set of available UAVs
Γ_k	Tripartite Chromosome of UAV k
d_{ij}, τ_{ij}	Distance and travel time for arc (i, j)
$q_i, [e_i, l_i]$	Demand and service time window for customer i
$Q, E_{\max}$	UAV capacity and maximum energy budget
μ, λ	Load-dependent energy coefficients
ω_1, ω_2	Objective weighting factors
x_{ij}^k	Binary: 1 if UAV k traverses arc (i, j)
t_i^k	Continuous: Arrival time of UAV k at node i
u_{ij}^k	Continuous: Payload carried by UAV k on arc (i, j)
b_i^k	Continuous: Remaining battery of UAV k at node i

A. Mathematical Model

At any decision epoch, the optimization targets the set of known active requests $\mathcal{V}^{\text{a}}$. The goal is to minimize the weighted sum of total fleet travel distance and time window violations, balancing efficiency with service quality:

$$\min Z = \omega_1 \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} d_{ij} x_{ij}^k + \omega_2 \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V}^{\text{a}}} \max(0, t_i^k - l_i) \tag{1}$$

Subject to the following constraints:

1) Routing and Flow Conservation: Standard VRP constraints ensure that each active customer is visited exactly once and flow continuity is maintained for every vehicle:

$$\sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{N}} x_{ij}^k = 1, \quad \forall i \in \mathcal{V}^{\mathrm{a}} \tag{2}$$

$$\sum_{j\in\mathcal{N}} x_{ij}^k - \sum_{j\in\mathcal{N}} x_{ji}^k = 0, \quad \forall k \in \mathcal{K}, i \in \mathcal{V}^{\mathrm{a}} \tag{3}$$

2) Payload Consistency: To accurately track load-dependent dynamics, we define u_{ij}^k as the payload carried during the flight on arc (i, j). Constraints (4)–(5) enforce flow conservation and capacity limits:

$$\sum_{j\in\mathcal{N}} u_{ji}^k - \sum_{j\in\mathcal{N}} u_{ij}^k = q_i \sum_{j\in\mathcal{N}} x_{ij}^k, \quad \forall i \in \mathcal{V}^{\mathrm{a}}, k \in \mathcal{K} \tag{4}$$

$$0 \le u_{ij}^k \le Q \cdot x_{ij}^k, \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K} \tag{5}$$

3) Physics-Aware Energy Feasibility: Constraint (6) recursively calculates energy depletion. A crucial feature is the term $(\mu + \lambda \cdot u_{ij}^k)$, which explicitly couples energy consumption to the variable payload u_{ij}^k, enforcing the load-dependent flight dynamics assumption:

$$b_j^k \le b_i^k - (\mu + \lambda \cdot u_{ij}^k) \cdot d_{ij} + M(1 - x_{ij}^k), \forall (i, j) \in \mathcal{A}, k \in \mathcal{K} \tag{6}$$

$$0 \le b_i^k \le E_{\max}, \quad \forall i \in \mathcal{N}, k \in \mathcal{K} \tag{7}$$

4) Time Window Propagation: Arrival times are propagated based on travel duration τ_{ij}. Soft time windows allow violations penalized in Z, the logical flow is governed by:

$$t_j^k \ge t_i^k + \tau_{ij} - M(1 - x_{ij}^k), \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K} \tag{8}$$

B. POMDP Formulation

Given the stochastic arrival of requests and the irreversibility of flight commitments, we formulate the D-ECVRPTW as a Partially Observable Markov Decision Process (POMDP). The decision process evolves over discrete epochs t, defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O}, \gamma \rangle$:

- **State ($\mathcal{S}$):** The latent system state s_t captures the full ground truth, including the active request set $\mathcal{V}^{\mathrm{a}}$, the fleet status $\mathbf{F}_t$, and the *unknown* future demand $\mathcal{V}^{\mathrm{p}}$.
- **Action ($\mathcal{A}$):** At each epoch, an action a_t updates the routing plan. The feasibility of a_t is constrained by the current vehicle locations and committed paths.
- **Transition ($\mathcal{T}$):** The system evolves based on the deterministic execution of routes and the stochastic realization of new demand from $\mathcal{V}^{\mathrm{p}}$.
- **Objective:** The fundamental goal is to minimize the cumulative operational cost over the horizon, to represent this objective in the POMDP, the reward is designed as the a differential improvement of the global objective value (Eq. 10)

This theoretical framework allows us to approach the dynamic routing problem as a sequential decision-making task. To tractably implement this POMDP, we bridge the gap between theoretical formulation and practical execution via a hierarchical architecture. The observation space Ω is projected into the tripartite chromosome segments (Γ_k) for causal consistency, while the action space $\mathcal{A}$ is explored through a coordinated manager-worker scheme. In the following section, we introduce HNES, which instantiates these POMDP elements into a closed-loop neuro-symbolic system designed for real-time responsiveness.

III. Methodology

To resolve the fundamental conflict between computational responsiveness and physical feasibility, we propose the Hybrid Neuro-Evolutionary Search (HNES) framework. As illustrated in Fig. 2, HNES is formulated as a neuro-symbolic cognitive architecture, integrating data-driven intuition with model-based reasoning. The foundation of the system is the state projection mechanism, which maps the stochastic environment into a tripartite chromosome (Γ_k). This structure strictly segregates the timeline into History, Locked Commitment, and Future Queue, creating a causal optimization manifold that underpins all decisions. Built upon this physics-aware representation, the control architecture operates on two distinct levels. First, the High-Level Meta-Control (The Manager) employs a DRL agent instantiated via an Actor-Critic architecture to orchestrate the system. Instead of optimizing routes directly, it learns a policy π_θ to dynamically arbitrate the timing of optimization based on state entropy. Second, the Low-Level Dual-Path Execution (The Workers) implements the Manager's directives by activating either a Neural Generator for global reconstruction or an Evolutionary Module for local constraint repair, as detailed in Section III-B.

A. Tripartite Chromosome: Modeling Irreversibility and Physics

Unlike static VRP where a solution is a simple spatial permutation, the D-ECVRPTW operates in a 'rolling horizon' where physical states are non-linear and temporally coupled. A viable encoding must reconcile three conflicting physical realities: (1) Causality: energy consumption is path-dependent, requiring a record of the past; (2) Kinetic Continuity: a drone in flight cannot instantly reroute without stability risks, necessitating a 'locked' commitment; and (3) Optimization Flexibility: the system must maintain a search space for future tasks. To internalize these constraints directly into the evolutionary search, we propose the Tripartite Chromosome Encoding (Fig. 3) where $\mathcal{H}_k$, $\mathcal{L}_k$, and $\mathcal{F}_k$ represent the Immutable History, Locked Commitment, and Future Queue, respectively. As illustrated in Figure 3, this structure serves not merely as a data container, but as a mechanism to enforce physical continuity constraints implicitly.

1) Immutable History ($\mathcal{H}_k$): The segment $\mathcal{H}_k$ records the sequence of nodes already visited by the UAV. In our physics-aware system, this segment functions as the Energy Anchor. Due to the Load-Dependent Energy Model (Eq. 6), the energy consumption of any flight segment is non-linear and depends on the cumulative payload history. Therefore, the current residual energy E_{res} cannot be determined in

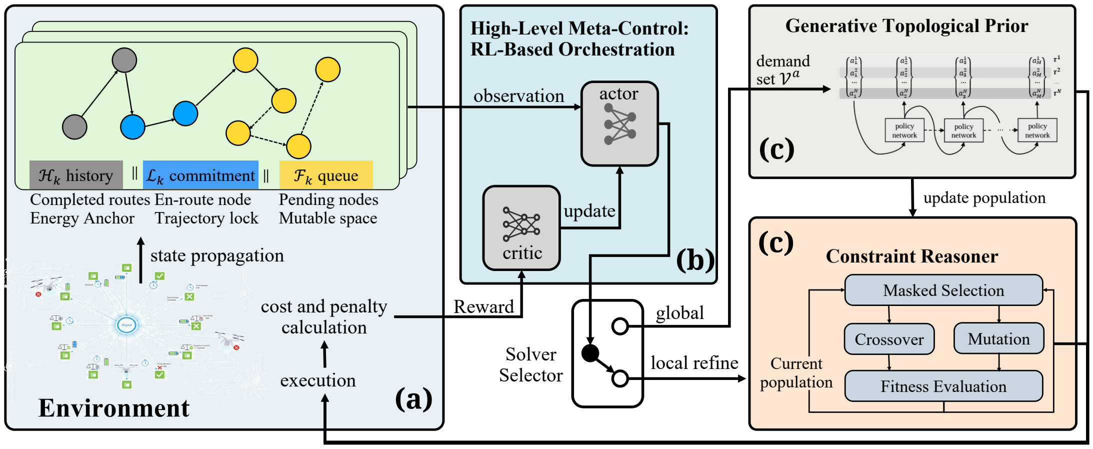

Fig. 2. **The Cognitive Architecture of HNES.** The framework operates as a neuro-symbolic cognitive system. **(A) State Projection:** The stochastic environment is projected into a tripartite state (Γ_k) to enforce causal consistency. **(B) Meta-Control:** The Hierarchical Manager (RL) extracts physics-aware features from these chromosome segments to orchestrate the solver hierarchy. **(C) Dual-Path Execution:** The generative topological prior [38]handles rapid structural adaptation, while the physics-informed constraint reasoner ensures rigorous feasibility.

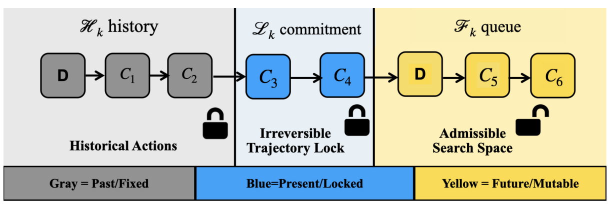

Fig. 3. The chromosome acts as a timeline. The History segment ($\mathcal{H}_k$) provides the integral basis for calculating load-dependent energy; the Locked segment ($\mathcal{L}_k$) represents the irreversible immediate trajectory; and the Future segment ($\mathcal{F}_k$) forms the constrained evolutionary search space.

isolation; it must be calculated by integrating the discharge rate over the trajectory defined by $\mathcal{H}_k$. From an evolutionary perspective, $\mathcal{H}_k$ is strictly "frozen"—protected from all genetic operators—ensuring that the fitness evaluation is grounded in the actual physical history of the flight.

2) Locked Commitment ($\mathcal{L}_k$): $\mathcal{L}_k$ represents the immediate destination the UAV is currently navigating towards. This segment functions as a Trajectory Lock. In real-world aerial logistics, redirecting a drone mid-flight introduces unacceptable control latency and stability risks. Thus, we treat the active edge as physically irreversible. Algorithmically, $\mathcal{L}_k$ acts as a fixed boundary condition. It serves as the transition interface that propagates the UAV's state (current location, time, and carried load) from the deterministic past to the stochastic future. Like $\mathcal{H}_k$, this segment is shielded from variation, preventing the algorithm from generating physically infeasible "teleportation" instructions.

3) Future Queue ($\mathcal{F}_k$): The final segment, $\mathcal{F}_k$, encapsulates the sequence of assigned but unexecuted tasks. Specifically, when new requests $\mathcal{V}^{\text{u}}$ (unassigned nodes) arrive, they are held in a global buffer. Upon the activation of an optimization action (a_{NEUR}), these nodes are integrated into the Future Queue $\mathcal{F}_k$ of the most suitable UAV. This constitutes the an Admissible Search Space. Unlike the rigid antecedent segments, $\mathcal{F}_k$ remains topologically plastic, serving as the designated buffer for accommodating dynamic insertions and sequence reordering. Mechanistically, the system enforces a strict State Propagation Protocol: the terminal boundary conditions of the Locked Commitment $\mathcal{L}_k$ (i.e., the projected time, location, and load upon arrival) serve as the immutable initialization state for $\mathcal{F}_k$. This isolation allows downstream solvers to freely explore combinatorial modifications—optimizing for energy efficiency and time-window adherence—within this future horizon, without violating the causality of historical actions or the stability of immediate operations.

B. High-Level Meta-Control: RL-Based Orchestration

The central challenge in dynamic logistics extends beyond solving a static VRP instance; it lies in mastering the Timing of Optimization—determining when and how to intervene in an evolving stochastic process. We position a Hierarchical Meta-Controller, tasked with learning an adaptive policy for algorithmic orchestration rather than direct node manipulation. Unlike traditional approaches that rely on rigid, periodic re-optimization, our agent actively diagnoses the system's state to arbitrate between conflicting operational needs.

To solve the POMDP formulated in Section II, the "Hierarchical Manager" learns a policy $\pi_\theta(a_t|o_t)$ to orchestrate the optimization strategy. We instantiate the specific MDP components as follows:

1) Observation Space: To facilitate efficient learning, we map the system state into a physics-aware observation space. We define the observation o_t as a composite structure containing both global demand statistics and vehicle-specific states:

$$o_t = [\mathbf{h}_t^{\text{DEMAND}} \parallel \mathbf{H}_t^{\text{FLEET}}] \tag{9}$$

where $\mathbf{h}_t^{\text{DEMAND}} = [\mu_c, \sigma_c, |\mathcal{V}^{\text{a}}|]$ represents demand entropy, quantifying the spatial centroid μ_c and dispersion σ_c of newly arrived requests. $\mathbf{H}_t^{\text{FLEET}} = \{\mathbf{v}_{k,t}\}_{k\in\mathcal{K}}$ is the fleet state tensor, where each UAV k is represented by a composite feature vector: $\mathbf{v}_{k,t} = [\Gamma_k \parallel b_{k,t}, \rho_{k,t}, \mathbf{x}_{k,t}]$ Here, Γ_k is the Tripartite Chromosome which provides the topological plan. $b_{k,t}$ is the residual energy; $\rho_{k,t}$ is current payload ratio projected from the active load in $\mathcal{L}_k$ and future demand in $\mathcal{F}_k$; $\mathbf{x}_{k,t}$ denotes the current position of the locked segment $\mathcal{L}_k$.

2) Discrete Meta-Action Space: The agent controls the optimization hierarchy via a discrete action space $\mathcal{A} = \{a_{\text{NEUR}}, a_{\text{EVO}}, a_{\text{NOOP}}\}$. Specifically, a_{NEUR} (Global Reconstruction) triggers the Neural Generator (System 1) to reset the mutable queue $\mathcal{F}_k$ from scratch, addressing high-entropy distributional shifts. Conversely, a_{EVO} (Evolutionary Refinement) invokes the Masked Evolutionary Operator (System 2) to iteratively repair hard physical constraints within the existing topology. Finally, a_{NOOP} (Maintain) preserves the current plan, allowing the system to conserve computational resources when the marginal utility of re-optimization is negligible.

3) Shaped Reward Function: The core challenge in dynamic VRP is balancing operational efficiency (low cost) with service responsiveness (low waiting time). If the agent only focuses on routing cost, it may delay assigning distant customers to avoid increasing the immediate path length. To counter this, we design a physics-aware shaped reward:

$$r_t = \underbrace{(Z(s_t) - Z(s_{t+1}))}_{\text{Cost Improvement}} - \beta \cdot \underbrace{\Phi(\mathcal{V}^{\text{u}}, \mathbf{w})}_{\text{Backlog Penalty}} \tag{10}$$

where $Z(s)$ represents the global objective value (Eq. 1). To ensure sustained responsiveness, we define the Backlog Penalty as $\Phi(\mathcal{V}^{\text{u}}, \mathbf{w}) = \sum_{j \in \mathcal{V}^{\text{u}}} (1 + w_j)^{\alpha}$, where w_j is the accumulated waiting time of unassigned request j and $\alpha \geq 1$ is a sensitivity coefficient. This penalty allows the agent to prioritize long-waiting requests to prevent customer "starvation." In this work, we adopt $\alpha = 1$.

4) Policy Optimization Algorithm: To stabilize the learning of this high-level policy within a non-stationary environment, we employ the Proximal Policy Optimization (PPO) algorithm. Standard policy gradient methods often suffer from high variance in dynamic routing scenarios, where a single poor decision can lead to irreversible mission failure. PPO mitigates this by limiting the magnitude of policy updates, ensuring monotonic improvement [39]. The agent optimizes the clipped surrogate objective:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(\rho_t(\theta) \hat{A}_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon) \hat{A}_t \right) \right] \tag{11}$$

where the probability ratio $\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ measures the deviation from the previous policy. The clipping function strictly bounds this ratio within $[1-\epsilon, 1+\epsilon]$, preventing destructive policy shifts and stabilizing the delicate interaction between the neural and evolutionary modules.

C. Generative Topological Prior

To overcome the inherent latency of evolutionary algorithms, we employ a pre-trained Transformer (based on POMO) as the framework's generative engine. The primary function of this module is not to perform final optimization, but to supply a generative topological prior. When triggered, the network exploits the rotational symmetry of VRP instances to generate N diverse trajectory skeletons in a single forward pass, providing 'data-driven intuition' that narrows the search space for the more computationally intensive evolutionary module. However, raw neural outputs are topologically agnostic to the UAV's real-time status. To bridge this gap, we implement a state-aware injection protocol: the linear sequences generated by the network are projected specifically onto the Future Queue ($\mathcal{F}_k$) segment of the chromosome. The system strictly respects the boundary conditions provided by the Locked Commitment ($\mathcal{L}_k$), ensuring continuity. This process transforms the neural network into a population seeder, injecting structural diversity into the search space effectively.

D. Physics-Informed Constraint Reasoner

While the neural prior provides a high-quality global skeleton, it lacks the inductive bias to guarantee feasibility under the non-linear energy dynamics of the D-ECVRPTW. Therefore, we position the evolutionary module as a physics-informed constraint reasoner, projecting neural approximations onto the feasible manifold through simulation-based validation. This module is grounded in the NSGA-II [40], a choice motivated by the inherent multi-objective nature of dynamic logistics—where the system must simultaneously minimize operational energy consumption and the non-linear backlog penalty. By maintaining a Pareto-optimal front, the reasoner prevents the search from collapsing into local optima that favor fuel efficiency at the expense of customer starvation.

To rigorously respect the causality of dynamic routing, we design a set of masked evolutionary operators. These operators utilize a binary mask to freeze the Immutable History ($\mathcal{H}_k$) and Locked Commitment ($\mathcal{L}_k$) segments, applying stochastic changes exclusively to the Future Queue ($\mathcal{F}_k$). Specifically, the masked load-balancing crossover targets energy heterogeneity across the fleet by transferring tasks from energy-critical UAVs to those with surplus endurance, ensuring that the committed locked segments remain untouched. Simultaneously, the masked temporal mutation focuses on resolving local time-window violations by applying heuristic reordering strategies within the mutable future segment. This module iteratively reasons about physical constraints, ensuring that every offspring generated is structurally valid and topologically optimized.

IV. Experimental Studies

A. Experimental Setup

1) Simulation Environment and Benchmarks: To generalize performance across varied topological structures, we utilize the Solomon Benchmark suite [42], expanded to instance sizes of $N \in \{100, 200, 400\}$ to rigorously assess scalability. The topological configurations are categorized into three distributions: Random (R) sampled from a uniform distribution $U[0,1]^2$ to represent unstructured environments; Clustered (C) mimicking high-density urban zones; and Mixed (RC) combining both patterns to ensure spatial robustness. To simulate evolving demand patterns within the D-ECVRPTW, we employ a Multi-Stage Dynamic Release Protocol. Each instance is divided into five euqal-sized subsets.The first segment functions as the static base known a priori at $t = 0$, while the remaining

TABLE III
COMPARATIVE PERFORMANCE EVALUATION ON SOLOMON BENCHMARKS. METRICS ARE ABBREVIATED AS: **T** (TIME), **D** (DISTANCE), **P** (TW PENALTY), AND **Z** (OBJECTIVE VALUE).

Size	**Inst.**	**ACS-KM** [28]				**reld-nco** [41]				**POMO** [38]				**NSGA-II** [16]				**Ours**			
		T	D	P	Z	T	D	P	Z	T	D	P	Z	T	D	P	Z	T	D	P	Z
100	C104	30s	1810	84	1894	1s	919	475	1394	1s	1612	128	1740	19s	1728	56	1784	15s	**1028**	**87**	**1115**
	C204	32s	1315	**0**	1315	1s	**813**	2492	3305	1s	2046	1	2047	24s	1208	314	1522	14s	1312	1	**1313**
	R104	32s	1146	**235**	1381	1s	1143	847	1990	1s	1189	400	1589	23s	1027	435	1472	16s	**938**	301	**1238**
	R204	30s	1485	**0**	1485	1s	**1034**	1154	2189	1s	1313	59	1372	22s	1205	230	1435	14s	1317	12	**1329**
	RC104	32s	1282	**184**	1466	1s	1297	665	1963	1s	1427	438	1865	22s	1355	324	1679	14s	**1172**	263	**1435**
	RC204	32s	2208	**0**	2208	1s	**1053**	842	1895	1s	1436	184	1620	20s	1225	236	1461	13s	1306	69	**1375**
200	C1_2_4	3m	4791	**702**	5493	3s	4590	2516	7106	2s	4459	949	5408	42s	3925	1346	5271	31s	**3636**	1063	**4699**
	C2_2_4	3m	4282	87	4369	3s	**3764**	5768	9532	3s	5318	171	5489	45s	4123	108	4236	28s	3995	**56**	**4051**
	R1_2_4	2.8m	3733	1472	5205	2s	4688	5156	9844	4s	**3503**	2338	5842	41s	3921	1730	5651	26s	3733	**1472**	**5205**
	R2_2_4	3.1m	5920	330	6250	4s	**3899**	18132	22031	4s	4972	**309**	5281	43s	6121	470	6591	35s	4024	574	**4598**
	RC1_2_4	3m	3523	1140	4663	5s	4229	4240	8469	3s	3680	2303	5983	41s	3461	1354	4815	32s	**3106**	**857**	**3963**
	RC2_2_4	2.9m	4724	**222**	4946	5s	3313	2697	6010	4s	3813	813	4626	43s	4516	367	4883	38s	**3727**	892	**4619**
400	C1_4_4	20m	9724	1323	11047	30s	17240	4801	22041	26s	**9721**	2370	12091	12m	9724	1323	11047	5m	9872	**1021**	**10893**
	C2_4_4	21m	15278	755	16033	35s	15455	6269	21724	30s	15036	**706**	15743	14m	16237	1072	17309	6m	**11947**	823	**12770**
	R1_4_4	22m	33149	**829**	33978	32s	15122	6921	22043	29s	**13904**	5273	19177	14m	35183	976	36159	5.7m	14583	1744	**16327**
	R2_4_4	21m	21633	1813	23446	32s	**16262**	6506	**22768**	31s	16902	7949	24851	14m	23591	2135	25726	5.8m	21633	**1813**	23446
	RC1_4_4	21m	**7022**	2384	9406	32s	11543	8965	20508	32s	9766	4690	14456	13m	7312	2145	9457	6m	7435	**1862**	**9297**
	RC2_4_4	22m	14746	**1602**	16348	32s	17565	12374	29939	30s	**9915**	12728	22643	14m	15036	1687	16723	6m	13844	1708	**15552**

are injected as dynamic waves at stochastic intervals. A strict Temporal Causality Constraint ensures every request is revealed prior to its time window opening, compelling the solver to execute adaptive re-optimization under sudden task influxes. Furthermore, to evaluate robustness under resource scarcity, we maintain a constrained fleet-to-task ratio. The specific fleet sizes ($|\mathcal{K}|$) assigned to different instance scales (N) are summarized in Table IV. This ensures that the solver is tested under high-utilization pressure rather than resource abundance.

TABLE IV
EXPERIMENTAL PARAMETER CONFIGURATIONS

Instance Scale (N)	**100**	**200**	**400**
Fleet Size ($\|\mathcal{K}\|$)	10	20	40
Avg. Requests per UAV	10	10	10

2) Implementation Details: The proposed framework was implemented in Python using the PyTorch library, executed on an Intel Core i9-11900 CPU with 64 GB RAM. The Neural Generator is instantiated as a Transformer-based encoder with $N = 6$ layers. The RL Manager is trained with PPO. We employ the Adam optimizer with decoupled learning rates for the actor ($\eta_a = 1 \times 10^{-4}$) and the critic ($\eta_c = 3 \times 10^{-4}$). To reduce variance in gradient estimation, we utilize Generalized Advantage Estimation (GAE) with $\lambda = 0.95$ and a discount factor $\gamma = 0.9$. Furthermore, an instance augmentation strategy (factor of 8) is applied during training to enhance sample efficiency by exploiting the geometric symmetries of VRP solutions. Finally, for the NSGA-II solver, the mutation and crossover rates are set to 0.1 and 0.7, respectively. The model was trained for 2000 eopchs on synthetic instances with problem scales ranging from 100 to 500 clients. To ensure generalization across diverse logistics scenarios, the training set follows three distinct spatial distributions: Random (R), Clustered (C), and Random-Clustered (RC), consistent with standard benchmark paradigms.

B. Comparative Performance Evaluation

Table III presents comprehensive statistical results on the Solomon Benchmarks with instance sizes $N \in \{100, 200, 400\}$, under the same constraints. The data reveals a distinct performance dichotomy among the baselines, which our framework successfully reconciles. Traditional heuristic methods prioritize feasibility but suffer from scalability issues. As shown in the $N = 200$ block, ACS-KM may achieves low penalties,but at the cost of excessive runtime (3 minutes) and suboptimal distances. Similarly, NSGA-II, while faster than ACS-KM, fails to converge to high-quality solutions in larger instances, yielding an objective value of $17,309$ on C2_4_4, which is significantly worse than HNES. Neural constructive approaches demonstrate superior inference speed and competitive distances in small instances. However, they exhibit a critical "Feasibility Collapse" as complexity scales. For example, on the $N = 200$ instance R2_2_4, reld-nco achieves a short distance but incurs a catastrophic time-window penalty, resulting in a non-competitive objective. This confirms that pure NCOs struggle to "deduce" strict physical constraints from data alone.

Our proposed HNES effectively bridges this gap. By seeding the evolutionary search with a high-quality neural prior, it avoids the "cold start" of NSGA-II; by enforcing physical constraints via the Tripartite Chromosome, it corrects the infeasibility of NCOs. Quantitatively, HNES consistently achieves the best overall objective. On large-scale instances, HNES outperforms the best baseline (ACS-KM or POMO) by margins ranging from 4% to 35%. Crucially, it maintains this quality within a reasonable computational budget, offering a Pareto-optimal balance for real-time dynamic logistics.

C. Ablation Studies: The Impact of Hybridization

To rigorously quantify the contribution of each module within the HNES framework, we conducted an ablation study isolating the generative engine and the constraint reasoner. We evaluate three distinct configurations: Generative-Only, where the solution is derived solely from the neural network in a single forward pass; Reasoner-Only, where the evolutionary module operates from a random initialization to isolate search capability without learned priors; and HNES (Full Hybrid), where the neural generator seeds the evolutionary reasoner.

The comparative results on instance RC1-2-4 are summarized in Table V. The Generative-Only baseline demonstrates superior computational speed ($T \approx 3$s) but suffers from high constraint violations (TW Penalty > 5000). This confirms that while the neural network captures general topological patterns, it fails to strictly deduce non-linear physical constraints for specific instances. Conversely, the Reasoner-Only configuration exhibits a severe "cold start" problem, requiring significant computational budget ($T \approx 3$m) to converge from random noise to a valid topology. The HNES configuration achieves the lowest objective value with minimized penalties in a fraction of the time. This validates the synergistic hypothesis: the neural module acts as a coarse-grained locator to bypass the expensive exploration phase, allowing the evolutionary module to function as a fine-grained tuner dedicated to rigorous physical feasibility.

TABLE V
ABLATION STUDY ON INSTANCE RC1-2-4 (AVG. OVER 10 RUNS)

Configuration	Obj. Value	TW Penalty	Time
Neural-Only	4724.7	5321.4	**3s**
Evo-Only	3859.2	1280.3	3m
HNES (Ours)	**3630.5**	**1140.4**	**27s**

D. Qualitative and Behavioral Analysis

To complement the quantitative results, we examine both route topology and policy behavior. Figure 4 shows that baseline methods produce more topological entanglement, with inefficient cross-cluster detours. In contrast, HNES forms a more coherent petaloid topology, reducing deadhead mileage, inter-route interference, and time-window penalty.

Figure 5 further shows a clear coarse-to-fine control pattern. The manager favors Global Reconstruction early, shifts toward Evolution Refinement in the middle stage, and increasingly selects Maintain near the deadline. This transition indicates that the policy adapts its computation budget to mission progress and preserves stability when large late-stage changes offer limited benefit.

V. CONCLUSION

In this paper, we addressed the trilemma of optimality, real-time responsiveness, and physical feasibility in the D-ECVRPTW. The proposed HNES combines a tripartite chromosome, an RL manager, and an evolutionary reasoner to preserve causal consistency while adapting online. Experiments on dynamic Solomon benchmarks show that HNES reduces cold-start effects, improves feasibility over pure neural methods, and achieves a strong balance between solution quality and runtime. Future work will consider stochastic factors such as wind and decentralized multi-agent extensions.

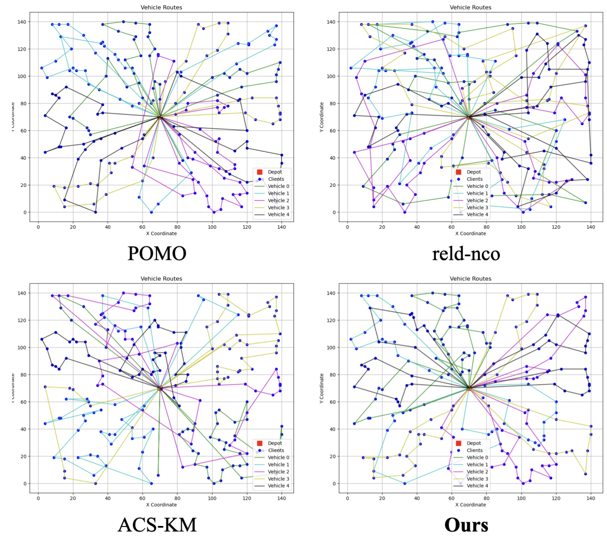

Fig. 4. Route visualization for instance R1-2-4 ($N = 200$). Our framework yields a more coherent topology with fewer inefficient detours and less inter-route interference than the baselines.

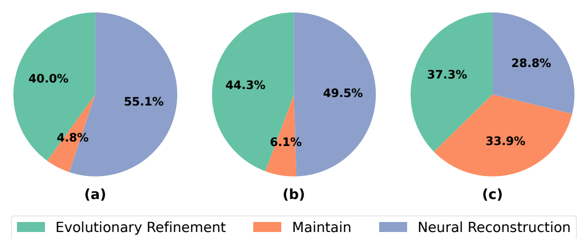

Fig. 5. Global action distribution over time (avg. of 60 instances from C, R, and RC). The policy shifts from early global construction **(a)** to mid-stage local repair **(b)**, and finally to a conservative strategy **(c)**.

REFERENCES

[1] T. Sheng, R. Jin, C. Yang, K. Qiu, M. Wang, J. Shi, J. Zhang, Y. Gao, Q. Wu, X. Zhou, H. Wang, J. Zhang, Q. Fang, N. Pan, Y. Xue, Y. Wang, R. Xiong, F. Gao, Y. Zhang, H. Lu, J. Yu, and Z. Gu, "Unmanned aerial vehicle mediated drug delivery for first aid," *Advanced Materials*, vol. 35, no. 10, p. 2208648, 2023.

[2] D. Edwards, N. Subramanian, A. Chaudhuri, P. Morlacchi, and W. Zeng, "Use of delivery drones for humanitarian operations: analysis of adoption barriers among logistics service providers from the technology acceptance model perspective," *Annals of Operations Research*, vol. 335, no. 3, pp. 1645–1667, Apr. 2024.

[3] F. Jia, Z. Zheng, C. Li, J. Xiao, R. Li, X. Yang, A. Rushworth, and S. Ijaz, "Design and validation of flexible aerial robotics for safe human-robot interaction," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 9342–9347.

[4] Z. Pei, T. Fang, K. Weng, and W. Yi, "Urban on-demand delivery via autonomous aerial mobility: Formulation and exact algorithm," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 3, pp. 1675–1689, 2023.

[5] I. Kucukoglu, R. Dewil, and D. Cattrysse, "The electric vehicle routing problem and its variations: A literature review," *Computers & Industrial Engineering*, vol. 161, p. 107650, 2021.

[6] J. Li, Y. Ma, R. Gao, Z. Cao, A. Lim, W. Song, and J. Zhang, "Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 13 572–13 585, Dec. 2022, publisher Copyright: © 2013 IEEE.

[7] A. N. Jasim and L. Chaari Fourati, "Guided genetic algorithm for solving capacitated vehicle routing problem with unmanned-aerial-vehicles," *IEEE Access*, vol. 12, pp. 106 333–106 358, 2024.

[8] Y. Li, M. Liu, and D. Jiang, "Application of unmanned aerial vehicles in logistics: A literature review," *Sustainability*, vol. 14, no. 21, 2022.

[9] J. Li, R. Liu, and R. Wang, "Elastic strategy-based adaptive genetic algorithm for solving dynamic vehicle routing problem with time windows," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 13 930–13 947, 2023.

[10] Y. Wang, Z. Wei, S. Luo, J. Zhou, and L. Zhen, "Collaboration and resource sharing in the multidepot time-dependent vehicle routing problem with time windows," *Transportation Research Part E: Logistics and Transportation Review*, vol. 192, p. 103798, 2024.

[11] O. S. da Silva Júnior, J. E. Leal, and M. Reimann, "A multiple ant colony system with random variable neighborhood descent for the dynamic vehicle routing problem with time windows," *Soft Computing*, vol. 25, no. 4, pp. 2935–2948, Feb. 2021.

[12] M. Tang, N. Yu, I. Karamouzas, and Z. Ye, "Terran: A transformer-based electric vehicle routing agent for real-time adaptive navigation," *IEEE Transactions on Automation Science and Engineering*, pp. 1–1, 2025.

[13] Z. Yuan, T. Wang, J. Tian, J. Zhang, J. Zheng, J. Wu, and Z. Gao, "Mitigate the range anxiety: two-stage optimization for the electric vehicle routing problem with time windows and battery status uncertainty," *Transportation Research Part E: Logistics and Transportation Review*, vol. 205, p. 104510, 2026.

[14] S. Meng, "The multi-visit parallel drone scheduling pickup and delivery problem considering multiple trips and time windows," *Transportation Research Part E: Logistics and Transportation Review*, vol. 203, p. 104351, 2025.

[15] S. Zhang, T. Zhou, C. Fang, and S. Yang, "A novel collaborative electric vehicle routing problem with multiple prioritized time windows and time-dependent hybrid recharging," *Expert Systems with Applications*, vol. 244, p. 122990, 2024.

[16] T. Hu, X. Yang, F. Jia, H. Xu, K. Yang, and T. Cui, "An enhanced multi-objective evolutionary algorithm for adaptive-formation multi-uav task allocation with load-dependent constraints," in *2025 IEEE Congress on Evolutionary Computation (CEC)*, 2025, pp. 1–8.

[17] Y. Hao, Z. Chen, X. Sun, and L. Tong, "Planning of truck platooning for road-network capacitated vehicle routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 194, p. 103898, 2025.

[18] C. Jin, R. Bai, Y. Zhou, X. Chen, and L. Tan, "Enhancing online yard crane scheduling through a two-stage rollout memetic genetic programming," *Memetic Computing*, vol. 16, no. 3, pp. 467–489, Sep. 2024.

[19] G. Ferro, M. Paolucci, and M. Robba, "Optimal charging and routing of electric vehicles with power constraints and time-of-use energy prices," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 436–14 447, 2020.

[20] C. Zhou, J. Ma, L. Douge, E. P. Chew, and L. H. Lee, "Reinforcement learning-based approach for dynamic vehicle routing problem with stochastic demand," *Computers & Industrial Engineering*, vol. 182, p. 109443, 2023.

[21] W. Pan and S. Q. Liu, "Deep reinforcement learning for the dynamic and uncertain vehicle routing problem," *Applied Intelligence*, vol. 53, no. 1, pp. 405–422, Jan. 2023.

[22] J. C. Pina-Pardo, D. F. Silva, A. E. Smith, and R. A. Gatica, "Dynamic vehicle routing problem with drone resupply for same-day delivery," *Transportation Research Part C: Emerging Technologies*, vol. 162, p. 104611, 2024.

[23] Z. Wang, R. Bai, F. Khan, E. Özcan, and T. Zhang, "Gase: graph attention sampling with edges fusion for solving vehicle routing problems," *Memetic Computing*, vol. 16, no. 3, pp. 337–353, Sep. 2024.

[24] B. Lin, B. Ghaddar, and J. Nathwani, "Deep reinforcement learning for the electric vehicle routing problem with time windows," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 528–11 538, 2022.

[25] G. Bono, J. S. Dibangoye, O. Simonin, L. Matignon, and F. Pereyron, "Solving multi-agent routing problems using deep attention mechanisms," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7804–7813, 2021.

[26] X. Yang, T. Cui, H. Wang, and Y. Ye, "Multiagent deep reinforcement learning for electric vehicle fast charging station pricing game in electricity-transportation nexus," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 4, pp. 6345–6355, 2024.

[27] X. Yang, Y. Gu, F. Jia, Y. Li, H. Wang, N. Du, T. Cui, Y. Ye, and R. Bai, "Evolution-assisted deep reinforcement learning for fast charging station coordinated operation," in *2024 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2024, pp. 1–9.

[28] Y. Teng, J. Chen, S. Zhang, J. Wang, and Z. Zhang, "Solving dynamic vehicle routing problem with time windows by ant colony system with bipartite graph matching," *Egyptian Informatics Journal*, vol. 25, p. 100421, 2024.

[29] Y. Zhang, R. Bai, R. Qu, C. Tu, and J. Jin, "A deep reinforcement learning based hyper-heuristic for combinatorial optimisation with uncertainties," *European Journal of Operational Research*, vol. 300, no. 2, pp. 418–427, 2022.

[30] J. Lin, X. Wang, R. Niu, and Y. He, "A q-learning-based hyper-heuristic for capacitated electric vehicle routing problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 10, pp. 15 746–15 757, 2025.

[31] T. Cui, X. Yang, F. Jia, J. Jin, Y. Ye, and R. Bai, "Mobile robot sequential decision making using a deep reinforcement learning hyper-heuristic approach," *Expert Systems with Applications*, vol. 257, p. 124959, 2024.

[32] S. Çelik, A. H. Schrotenboer, L. Martin, and T. Van Woensel, "Is waiting worth it? the value of delaying time window assignment in vehicle routing problems," *Transportation Research Part B: Methodological*, vol. 204, p. 103381, 2026.

[33] G. Tresca, H. Salem, G. Cavone, H. Zgaya-Biau, S. Ben-Othman, S. Hammadi, and M. Dotoli, "A matheuristic approach for delivery planning and dynamic vehicle routing in logistics 4.0," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 3345–3365, 2025.

[34] M. Mavrovouniotis, C. Menelaou, S. Timotheou, G. Ellinas, C. Panayiotou, and M. Polycarpou, "A benchmark test suite for the electric capacitated vehicle routing problem," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8.

[35] C.-T. Feng, Y.-H. Jia, Q. Yang, W.-N. Chen, and H. Jiang, "A bilevel hybrid genetic algorithm for capacitated electric vehicle routing problem," in *2024 IEEE Congress on Evolutionary Computation (CEC)*, 2024, pp. 1–8.

[36] S. Wang, W. Sun, and M. Huang, "An adaptive large neighborhood search for the multi-depot dynamic vehicle routing problem with time windows," *Computers & Industrial Engineering*, vol. 191, p. 110122, 2024.

[37] N. T. Tam, N. D. Anh, D. V. Tung, B. H. T. Thanh, and L. T. Vinh, "Evolving routing and sequencing policies for dynamic vehicle routing problem with time windows," *Applied Soft Computing*, vol. 189, p. 114234, 2026.

[38] Y.-D. Kwon, J. Choo, B. Kim, I. Yoon, Y. Gwon, and S. Min, "Pomo: Policy optimization with multiple optima for reinforcement learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 21 188–21 198.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.

[40] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[41] Z. Huang, J. Zhou, Z. Cao, and Y. XU, "Rethinking light decoder-based solvers for vehicle routing problems," in *International Conference on Learning Representations*, Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, Eds., vol. 2025, 2025, pp. 27 471–27 491.

[42] W. Zhao, R. Zhang, K. Luo, and W. Wang, "A two-stage stochastic programming method for a heterogeneous vehicle routing problem with time windows and stochastic demand," *Expert Systems with Applications*, vol. 291, p. 128463, 2025.