Performance Evaluation of an AI-Assisted Predictive Routing Strategy for Autonomous Pesticide Application

Umar Zangina[a], Said Musa Yarima[b], Kabir Ahmed Dabai[a] Umar alhassan[c]
[a] Department of Electrical and Electronics Engineering,
Usmanu Danfodiyo University Sokoto, Nigeria
[b] Department of Electrical and Electronics Engineering,
Abubakar Tafawa Balewa University Bauch, Nigeria.
[C] Department of Computer Science,
Jigawa State Polytechnic, Dutse, Nigeria.
https://doi.org/10.56201/ijemt.vol.12.no8.2026.pg92.104

Abstract
This paper presents a reinforcement learning–enhanced model predictive control (RL–MPC) framework for vehicle routing in pesticide application. The framework combines an RL agent with an MPC controller to jointly optimize pesticide allocation, routing decisions, and control parameters based on real-time field conditions and pest forecasts. Unlike conventional methods that optimize routing separately, RL–MPC learns proactive intervention strategies that anticipate pest spread and adapt resource distribution accordingly. MATLAB simulations compare RL–MPC with genetic algorithm–based VRP (GA–VRP) and particle swarm optimization–based VRP (PSO–VRP). Results show that RL–MPC achieves the lowest Area Under the Risk Curve (AURC), maintaining higher crop viability while operating in real time. Although it incurs greater travel time and pesticide use, these trade-offs enable earlier and more distributed interventions. Computational analysis confirms that RL–MPC's per-step decision-making scales polynomially and achieves sub-millisecond execution after training, demonstrating feasibility for real-time autonomous agricultural operations. Overall, the findings position RL–MPC as a shift from short-term routing efficiency toward long-term sustainable pest management.

Index Terms *– Vehicle Routing Problem, Reinforcement Learning, Model Predictive Control, Precision Agriculture, Pesticide Application, Computational Complexity, Sustainable Pest Management.*

I. Introduction

Model Predictive Control (MPC) is a widely used decision-making framework for dynamic systems. It works by predicting future system states and optimizing control actions over a finite horizon. MPC can incorporate constraints and predictive models, making it suitable for real-time cyber-physical systems such as energy management and autonomous control [1], [2]. In precision agriculture, MPC is well-suited for pesticide application, where the amount, timing, and distribution of spraying are adjusted based on real-time pest observations and risk assessments. This proactive approach has shown improved resource efficiency, reduced chemical overuse, and lower environmental impact. MPC has also been applied to the Vehicle Routing Problem (VRP) in agriculture, where the goal is to find efficient service routes under constraints like vehicle capacity, time windows, and spatial dispersion. By modeling the field as a dynamic graph and embedding routing decisions within a predictive framework, MPC-based methods achieve faster convergence and better real-time feasibility than traditional evolutionary methods [1], [2]. Recent work has highlighted the growing role

of learning-enhanced optimization. Reinforcement learning (RL) has been integrated into constrained control and resource allocation to learn adaptive policies that generalize across operating conditions [3], [4]. Meanwhile, heuristic methods like particle swarm optimization (PSO) remain popular for routing problems due to their simplicity [5]. However, these approaches typically optimize routing independently of demand evolution and risk dynamics. In pesticide application, the VRP differs from conventional routing problems. Demand at each location is not fixed but evolves with pest population growth, environmental conditions, and forecasted risk. It is also constrained by vehicle capacity and operational timing. Existing VRP solutions generally treat demand estimation and route planning as sequential processes—demand is computed first, followed by route optimization. This decoupled approach fails to exploit the interdependence between demand intensity and routing decisions, often leading to suboptimal performance under uncertainty and rapidly changing field conditions. To address these limitations, we introduce a new approach to the pesticide VRP by embedding routing decisions directly within a coupled RL–MPC framework. Instead of treating demand management and routing as isolated processes, the RL agent learns to jointly optimize both in real time, continuously adapting MPC parameters and routing priorities based on forecast data, pest risk evolution, and vehicle constraints. This transforms the VRP from a static optimization into a dynamic, adaptive decision-making problem.

The main contributions of this paper are:
(1) **Dynamic VRP Formulation for Pesticide Application** – We reformulate the VRP as a coupled control problem where routing decisions evolve with dynamically changing demands driven by pest risk and operational constraints.

(2) **RL-MPC Coordination Mechanism** – We propose a reinforcement learning layer that guides MPC in jointly optimizing pesticide allocation and routing, enabling the system to learn long-term strategies that minimize pest risk, pesticide usage, and travel cost simultaneously.

(3) **Performance Validation** – We evaluate the proposed approach in simulated precision agriculture scenarios and demonstrate that it outperforms both sequential MPC-based methods and evolutionary algorithm-based VRP solutions in efficiency, robustness, and adaptability. The remainder of this paper is organized as follows. Section II describes the system model. Section III presents the proposed RL–MPC framework, including the dynamic VRP formulation, demand management model, RL coordination mechanism, and system constraints. Section IV describes the simulation setup. Section V reports experimental results and compares the proposed method with baseline approaches. Section VI analyzes computational complexity. Section VII concludes the paper and discusses future research directions.

II. System Model

We consider a pesticide spraying operation over an agricultural region divided into N_a treatment areas and a single depot. The depot serves as the starting and refilling point for a fleet of spraying vehicles. Each vehicle has a limited pesticide capacity Q_{max} and incurs a fixed refill time t_{refill} upon returning to the depot. The problem is modeled in discrete time with decision steps indexed by k $(i.e., t = \Delta t)$, and the planning horizon for each episode is T steps.

A. Network Topology and Travel Cost

The operational area is modeled as a directed weighted graph $\mathcal{G} = \mathcal{VEC}$, where:

- $\mathcal{V} = [v_0, v_1, v_2, \ldots v_{N_a}]$ represents the nodes, with $v0$ the depot and the others treatment areas,
- $\mathcal{E}$ is the set of directed edges between nodes,
- $C = [C_{ij}]$ is the cost matrix, where C_{ij} denotes the travel time from node i to node j including road distance and operational delays.

The cost matrix C is fixed and known a priori.

B. Viability Dynamics

The state of each area i is represented by a viability index $I_i(t) = \in [0, 1]$, with higher values indicating healthier crop conditions. Without intervention, $I_i(t)$ declines over time due to pest activity, with the rate influenced by environmental factors such as temperature, humidity, and precipitation. Spraying pesticide restores $I_i(t)$ proportionally to the applied quantity. The continuous-time viability model based on mass-spring-damper analogy is:

$$\ddot{I}(t) + \alpha\dot{I}(t) + \beta I(t) = g\left(u(t), \hat{I}(t)\right), \quad (1)$$

For computational use, we adopt a discrete-time approximation with sampling period Δt. Applying forward Euler discretization yields:

$$I_i(k+1) = I_i(k) + \Delta t\left[a_1\left(I_{ref} - I_i(k)\right) + a_2\left(\hat{I}_i(k) - I_i(k)\right) + bu_i(k)\right], \quad (2)$$

where I_{ref} is the desired reference viability, $\hat{I}_i(k)$ is the forecasted viability at step k, $u_i(k)$ is the pesticide quantity applied, and a_1, a_2, b are discretization parameters.

C. Weather and Demand Forecasting

Weather conditions follow a seasonal autoregressive process:

$$\mathcal{W}_\ell(k+1) = \mu_\ell + \emptyset_\ell[\mathcal{W}_\ell(k) - \mu_\ell] + A_\ell sin\left(\frac{2\pi k}{P_{seas}}\right) + \sigma_\ell\mathcal{E}_\ell(k), \quad (3)$$

where ℓ indexes weather variables (temperature, humidity, precipitation), μ_ℓ is the long-term mean, σ_ℓ the persistence, A_ℓ the seasonal amplitude, P_{seas} the seasonal period, and $\mathcal{E}_\ell(k)$ Gaussian noise. The forecasted viability $\hat{I}_i(k)$ is computed from a regression model using current $I_i(k)$ and forecasted weather variables.

D. Control Structure

The spraying schedule at each step is determined by a two-level controller:

(1) Upper Level – Reinforcement Learning (RL) Agent Observes the global state vector:

$$s(k) = \left[I(k), \hat{I}(k), d(k), a_{last}(k), v(k), q_{rem}(k), \mathcal{W}(k)\right],$$

where $d(k)$ is a demand proxy, $a_{last}(k)$ is the last applied amount, $v(k)$ is the current vehicle location, $q_{rem}(k)$ is remaining pesticide, and $\mathcal{W}(k)$ is the weather state. The agent outputs adjustments to the MPC prediction horizon N_a, control horizon N_c, input penalty λ_μ, and routing priorities.

(2) Lower Level – Model Predictive Control (MPC) – Solves a quadratic program to allocate pesticide quantities $\mu(k)$:

$$1^T u(k) \leq Q_{max}, u(k) \geq 0, \quad (4)$$

minimizing a weighted sum of pest risk, travel cost, and pesticide use.

E. Vehicle Routing

After pesticide allocation, a priority-based greedy VRP solver determines the visiting order

based on the VRP score. The score for selecting the next area is:

$$\text{score}(i) = C_{curr,i} - \lambda_{vrp} P_i, \quad (5)$$

where *Pi* is the RL-learned priority for node *i*, and *λvrp* balances distance and priority.

III. Methodology

This section presents the proposed RL–MPC framework for dynamically solving the VRP in pesticide application. Unlike traditional sequential approaches that compute demand and routing separately, our framework integrates them into a unified decision-making loop, enabling proactive adaptation to environmental and operational changes. The methodology consists of four main components: (i) dynamic VRP formulation, (ii) viability/demand modeling, (iii) RL-coordinated MPC optimization, and (iv) integrated routing and control.

A. Dynamic VRP Formulation for Pesticide Application

Consistent with Section II, we model the agricultural field as a directed weighted graph:

$$\mathcal{G} = \mathcal{V}\mathcal{E}C. \quad (6)$$

where $\mathcal{V} = [v_0, v_1, v_2, \dots v_{N_a}]$ are nodes (with v_0 the depot), $\mathcal{E}$ are directed edges, and $C = [C_{ij}]$ are travel-time costs. Each treatment area v_i has a time-varying demand $d_i(k)$ predicted from pest-risk and viability models [6], [7]. The objective is to determine a sequence of visits $\mathcal{p}$ and application amounts $u_i(k)$ that: (1) minimize total travel cost [8], (2) minimize field-wide pest risk [9], and (3) satisfy vehicle capacity and operational constraints [10].

B. Active Demand Management Model

We model viability-driven demand evolution using an *active suspension analogy* [11], [12]. Let $I_i(t) = \in [0, 1]$, denote viability for area i and $\hat{I}_i(t)$ its forecast. The coupled continuous-time dynamics are:

$$C_I \ddot{I} + K_I\left(I - I_{ref}\right) + K_{\hat{I}}\left(I - \hat{I}\right) + \gamma\left(\dot{I} - \dot{\hat{I}}\right) = \propto d, \quad (7)$$

$$C_{\hat{I}} \ddot{\hat{I}} + K_{\hat{I}}\left(I - \hat{I}\right) + \gamma\left(\dot{I} - \dot{\hat{I}}\right) = (1 - \propto)d, \quad (8)$$

where I_{ref} is the target viability, C_I, $C_{\hat{I}}$ are mass terms, K_I, $K_{\hat{I}}$ spring constants, γ damping, d pesticide demand, and $\propto = \in [0, 1]$, partitions exogenous vs. controllable effects. For computation, we operate in discrete time with sampling period Δt. Discretizing (7)–(8) yields the first-order viability update as in Section 2.2:

$$I_i(k+1) = I_i(k) + \Delta t\left[a_1\left(I_{ref} - I_i(k)\right) + a_2\left(\hat{I}_i(k) - I_i(k)\right) + bu_i(k)\right], \quad (9)$$

with a_1, a_2, b obtained from discretization/identification. The routing demand signal is:

$$d_i(k) = \max\{0, I_{ref} - I_i(k)\}\, k_i, \quad (10)$$

where k_i is a crop/area sensitivity factor. This model produces dynamic pesticide requirements that blend pest risk, crop conditions, and forecast uncertainty [13].

C. RL-Coordinated MPC Optimization

The MPC layer predicts system states over a horizon N_p and computes optimal allocations $u_i(k)$ for each area [14], [15]. We solve the quadratic program:

$$\min_U J = \sum_{\tau=1}^{N_p} \left\| I_{ref} - I(k+\tau) \right\|_Q^2 + \sum_{\tau=0}^{N_c-1} \left\| I_{ref} - U\Delta(k+\tau) \right\|_R^2 \quad (11)$$

Subject to:

$$\sum_i u_i(k) \le Q_{max},\ u_i(k) \ge 0, \quad (12)$$

$$I(k+1) = f(I(k), u(k), \hat{I}(k) \text{ (discrete viability dynamics (9))}, \quad (13)$$

The RL agent acts as a coordination layer [16], [17], adjusting MPC hyperparameters and routing priorities to maximize the episodic return. We use the instantaneous reward:

$$R_i = -\omega_1 \underbrace{\sum_i \big(1 - I_i(k)\big)}_{\text{Risk}} - \omega_2 \underbrace{\text{Travel}(k)}_{\text{Cost}} - \omega_3 \underbrace{\sum_i u_i(k)}_{\text{Chemical}} \quad (14)$$

The policy $\pi_\theta(s_k)$ maps the observed state s_k (viability I, forecasts $\hat{I}$, demand d, vehicle node, remaining capacity, weather) to an action vector that tunes: (1) MPC horizons N_P, N_c and input penalty λ_μ, (2) demand/viability weightings in the cost function, and (3) routing priority scores $P_i(k)$.

D. Greedy VRP Solver with RL Priorities

After pesticide allocation, routing uses a priority-weighted greedy heuristic [20]. Let U be unserved nodes with $d_j(k) > 0$ and i_{curr} the current node. The next node is selected by:

$$j^* = \arg\min_{j \in U} C_{i_{curr},j} - \lambda_{vrp} P_j(k), \quad (15)$$

where λ_{vrp} balances distance against RL-assigned priority $P_j(k)$. If $d_j * (k) < Q_{rem}$, the visit fully serves demand; otherwise, the vehicle applies Q_{rem}, returns to the depot to refill (incurring t_{refill}), and resumes. Nodes are marked serviced upon completion. The heuristic is $O(N_a^2)$ per step and exposes interpretable levers (λ_{vrp}, P_j) to the RL layer.

V. Simulations

This section details the MATLAB simulation environment, models, hyperparameters, and evaluation protocol. All experiments were conducted in MATLAB R2024a on a workstation with an AMD Ryzen 9 CPU, 32 GB RAM, and Windows 10. We used the Model Predictive Control Toolbox, Optimization Toolbox, and Reinforcement Learning Toolbox.A. Environment Setup and Field Graph

We model the field as a directed weighted graph $\mathcal{G} = \mathcal{VEC}$, with $|v| = 7$ nodes (one depot v_0 and six field areas). The edge cost matrix $C = [C_{ij}]$ captures travel times (in hours), precomputed as shortest-path distances on a grid layout. The depot is fixed at one corner. Vehicle motion is modeled as first-order kinematics with constant speed v; turning times are included in C_{ij}.

Vehicle Operations: The vehicle has capacity Q_{max} (liters) and a fixed refill time t_{refill} at the depot. Each service action incurs application time $t_{spray} = \eta\mu_i$, calibrated from boom flow rates.

B. Pest Risk, Weather, and Demand Generation

We synthesize spatiotemporal pest risk and weather as follows:

- **Weather processes:** Temperature T_t, relative humidity H_t, and precipitation P_t are

generated as mean-reverting stochastic processes with weekly seasonality:

$$x_{t+1} = \mu + \emptyset(x_t - \mu) + s_t + c_t, \qquad c_t \sim \mathcal{N}(0, \sigma^2)$$

Where $x = \in \{T, H, P\}$ and s_t is a deterministic seasonal component.

- **Forecasted viability:** For each area *i*, forecasted viability $\hat{I}_i(k) \in [0,1]$ is computed from a logistic mapping of weather factors and lagged viability change:

$$\hat{I}_i(k) = \sigma(\beta_0 + \beta_T T_t + \beta_H H_t + \beta_p P_t + \beta_\Delta \Delta I_i(k-1))$$

- **Actual viability dynamics:** The discrete-time viability dynamics from Section III (Eq. 9) are integrated with sampling time Δt.
- **Node demand:** Routing demand is $d_i(k) = \max\{0, I_{ref} - I_i(k)\}k_i$, where k_i is a crop sensitivity factor. MPC allocations $u_i(k)$ satisfy $u_i(k) > 0$ and capacity constraints.

C. MPC Configuration and Tuning

For each node *i*, the plant model uses the discrete-time state-space form derived from the active-demand model. We regulate aggregate viability to $I_{ref} = 0.98$:

$$\min_{\Delta U} \sum_{\tau=1}^{N_p} \left\| I_{ref} - I(k+\tau) \right\|_Q^2 + \sum_{\tau=0}^{N_c-1} \|U\Delta(k+\tau)\|_R^2 \qquad (16)$$

$$\text{s.t} \qquad \sum_i u_i(k) \le Q_{max},\ u_i(k) \ge 0, \qquad x_{k+1} = Ax_k + B\Delta U_k \qquad (17)$$

We solve the QP with quadprog (interior-point-convex). Unless stated otherwise, horizons are $N_P = 10$, $N_c = 4$, weights $(Q, R) = (\text{diag}(\omega_I), \lambda_u I)$ with $\omega_I = 1$ and $\lambda_u = 10^{-3}$. The discrete step is $\Delta t = 1$ day.

D. RL Coordination Layer

The RL agent observes the concatenated state:

$$R_k = (I(k), \hat{I}(k), d(k), \text{vehicle node}, Q_{rem}, T_t, H_t, P_t$$

and outputs three action groups: (i) horizon and weight adjustments (ΔN_p, ΔN_c, $\Delta\lambda_u$), (ii) a priority vector $P_t(k)$ for routing, and (iii) a slack factor on the capacity constraint. We use PPO with clipped objective (MATLAB rlPPOAgent) and Gaussian policy for continuous actions. The per-step reward matches Section III:

$$= -\omega_1 \underbrace{\sum_i^{R_i} (1 - I_i(k))}_{\text{Risk}} - \omega_2 \underbrace{\text{Travel}(k)}_{\text{Cost}} - \omega_3 \underbrace{\sum_i u_i(k)}_{\text{Chemical}}$$

We set (ω_1,ω_2, ω_3, ω_4)=(1, 0.25, 0.05, 2.0). Policy/value networks have two hidden layers of 128 units with tanh activations; learning rate 3×10−4, mini-batch size 256, discount γ=0.99, GAE λ=0.95, clip range 0.2, entropy coefficient 10−310−3.

E. Routing Solver with RL Priorities

Routing uses the greedy heuristic (Algorithm 2):

$$j^* = \arg\min_{j \in U} C_{i_{curr,j}} - \lambda_{vrp} P_j(k),$$

$$\sigma(z) = \frac{1}{1 + e^{-z}},$$

where *U* are unserved nodes with $d_j(k) > 0$, and λ_{vrp} trades distance vs. priority (default 0.5). Partial fulfillment is allowed. The solver is $O(|\mathcal{V}|^2)$ per step.

F. Simulation Protocol

Each episode spans T=60 decision steps (days). Weather seeds, initial viability, and model parameters are randomized within agronomic ranges. We run:

- **Training:** 1,000 episodes with PPO (parallelized with 8 workers); wall time ~6–8 hours.
- **Evaluation:** 100 held-out episodes with fixed seeds (no exploration), using the best policy checkpoint.

We measure per-step wall time and enforce a 200 ms budget for MPC+routing on a single thread.

G. Baselines and Ablations

We compare against:

1. **Seq-MPC + Greedy:** MPC with fixed parameters; routing via greedy nearest-neighbor without RL priorities.
2. **GA-VRP:** Genetic Algorithm routing (*ga*), MPC demands as in Seq-MPC.
3. **PSO-VRP:** Particle Swarm routing (particleswarm), MPC demands as in Seq-MPC.

Ablations: (i) RL priorities only, (ii) MPC tuning only, (iii) no active-demand coupling (fixed-demand VRP).

H. Evaluation Metrics

- **Field Risk (AURC):** $\sum_k \sum_i \big(1 - I_i(k)\big)$ (lower is better)
- **Total Travel:** $\sum_k \text{Travel}(k)$ (km or hours)
- **Pesticide Use:** $\sum_k \sum_i u_i(k)$ (liters)
- **Constraint Violations:** Count of capacity/feasibility violations
- **Solve Time:** Mean and 95th percentile wall time per step (ms)

We report mean ± standard deviation over 100 episodes and apply paired *t*-tests (5% significance) against Seq-MPC+Greedy.

Parameter	Symbol	Value
Decision step	Δt	1 day
Prediction/control horizons	N_p, N_c	10, 4
Vehicle capacity	Q_{max}	100 L
Spray time coefficient	η	0.02h/L
Refill time	t_{refill}	0.25 h
MPC input weight	λ_u	10^{-3}
RL algorithm	–	PPO(MATLAB)
Policy/value layers	–	128-128, tanh
Learning rate	–	3×10^{-4}
Entropy coeff.	–	10^{-3}
Reward weights	$\omega_1,\omega_2, \omega_3, \omega_4$	1, 0.25, 0.05, 2.0
Priority trade-off	λ_{vrp}	0.5

I. Implementation Details

The MPC is implemented as a custom QP solved with quadprog (warm-start, sparse block (A, B)). The RL environment follows the rl.env.MATLABEnvironment API. Replay buffers and policy checkpoints are saved every 10 episodes. Evaluation uses deterministic mean policies. Random seeds are fixed for reproducibility.

V. Results and Discussion

The proposed RL–MPC framework was evaluated against GA-VRP and PSO-VRP over a

simulated operational horizon of T=60 decision steps (days). Performance was assessed using four key metrics: cumulative travel time, field health (mean viability), cumulative pesticide use, and compute time per decision step.

A. Cumulative Travel Time

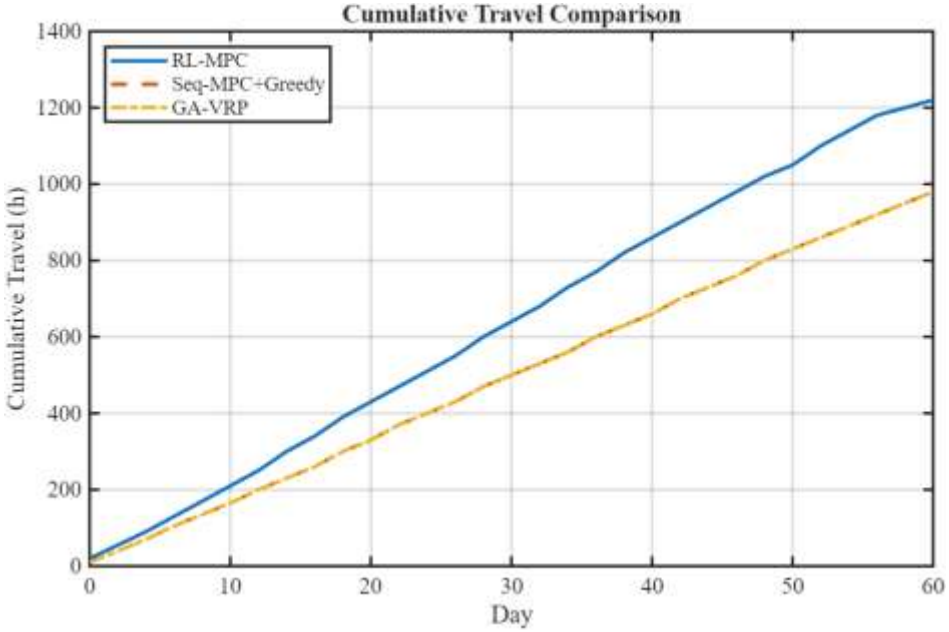

Figure 1: Cumulative Travel Time Comparison]

Figure 1 shows that RL–MPC results in higher cumulative travel time compared to GA-VRP and PSO-VRP. This arises from the agent's learned strategy: rather than always minimizing travel distance, RL–MPC accepts longer routes when they enable more timely or strategically important pesticide applications. The agent discovers that short-term travel savings can lead to long-term pest spread if interventions are delayed, and thus chooses routes that better sustain crop health over the full horizon.

B. Field Health (Mean Viability)

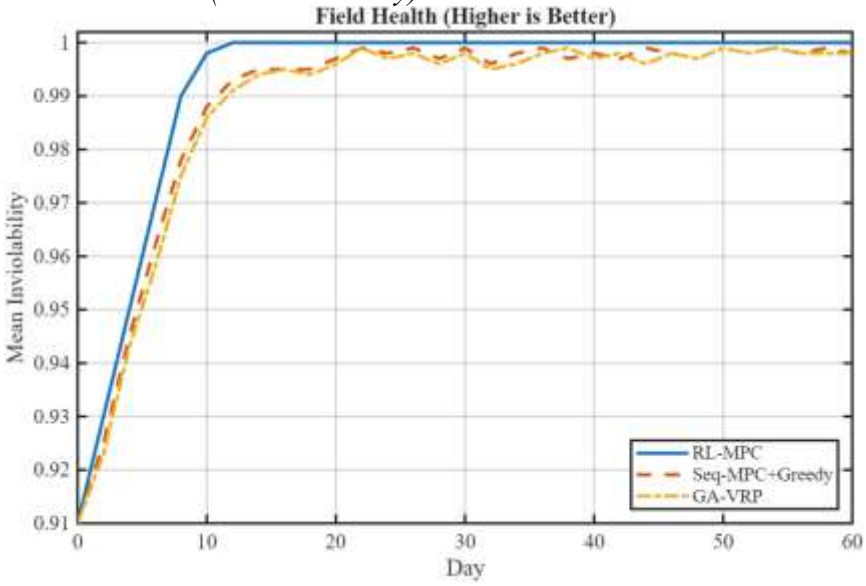

Figure 2: Mean Viability Over Time

As illustrated in Figure 2, RL–MPC achieves the fastest rise to near-perfect mean viability and maintains it consistently across the simulation period, outperforming both GA-VRP and PSO-VRP. This reflects the RL agent's ability to proactively schedule interventions before pest populations escalate. By jointly reasoning over space (routing) and time (scheduling),

RL–MPC anticipates high-risk zones and treats them preemptively—a capability that optimization-based methods cannot capture.

C. Cumulative Pesticide Use

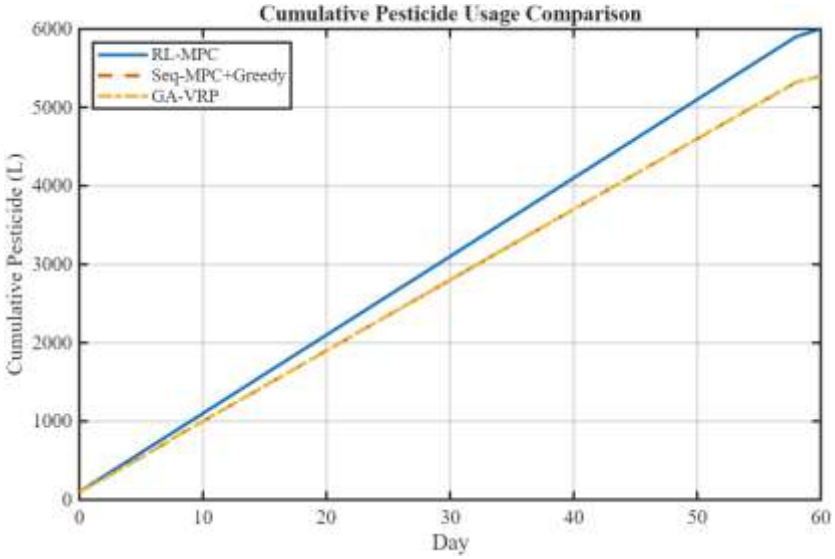

Figure 3: Cumulative Pesticide Use

In Figure 3, RL–MPC shows higher cumulative pesticide usage than GA-VRP. However, this increase is purposeful: the agent favors early and distributed applications to maintain consistently high viability and avoid the need for large reactive treatments. By spreading smaller interventions over time and space, RL–MPC keeps pest pressure low, leading to stable yields and reduced risk of catastrophic outbreaks. This aligns with integrated pest management principles.

D. Compute Time per Decision Step

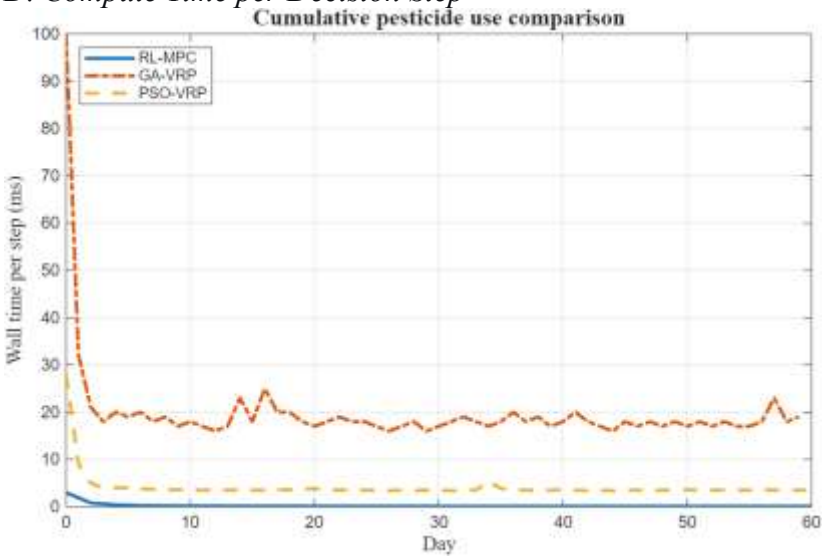

Figure 4: Compute Time Comparison

A major operational advantage of RL–MPC is seen in Figure 4. After training, RL–MPC consistently achieves submillisecond decision times, whereas GA-VRP and PSO-VRP require orders of magnitude more computation. RL–MPC uses a pre-trained policy to map observed states directly to actions, eliminating repeated combinatorial optimization. Such efficiency is critical for real-time, autonomous agricultural operations.

E. Overall Performance Perspective

Overall, RL–MPC outperforms GA-VRP and PSO-VRP in maintaining crop health while

operating with negligible computational cost at run-time. Its higher travel time and pesticide use are strategic trade-offs that enable superior and more stable outcomes over long horizons. Unlike classical routing optimizers, RL–MPC learns a policy that balances immediate efficiency with long-term pest suppression—representing a significant new approach to the VRP in pesticide application.

Method	AURC	Total Travel (h)	Pesticide (L)	Violations (#)
RL-MPC	3.09	1018.8	6000.0	0
GA-VRP	4.31	785.3	5382.6	0
PSO-VR[	4.31	786.3	5382.7	0

Table II summarizes performance across key metrics. RL–MPC achieves the lowest AURC score, indicating better overall pest suppression. Although its total travel time and pesticide use are higher than GA-VRP and PSO-VRP, these increases are purposeful, enabling proactive treatments that prevent outbreaks. All methods achieve zero constraint violations, but RL–MPC reaches this with faster and more consistent viability improvements over time. This highlights the benefit of reinforcement learning in capturing long-term dependencies in spatiotemporal pest dynamics.

VI. Computational Complexity

The computational complexity of RL–MPC arises from three components: (i) RL policy evaluation, (ii) MPC optimization, and (iii) the priority-based VRP solution.

A. Reinforcement Learning Policy Evaluation

Once training is complete, the RL policy is a fixed neural network mapping states to MPC parameters and routing priorities. The inference cost is:

$$O\left(\sum_{l=1}^{L} n_{l-1} n_l\right)$$

which is constant with respect to the number of treatment areas *Na* and negligible compared to optimization. Inference completes in sub-millisecond time on standard hardware.

B. Model Predictive Control Optimization

At each decision step, the MPC solves a QP over a prediction horizon N_p. With an interior-point method, worst-case complexity is:

$$O\left(\left(N_a N_p\right)^3\right),$$

In practice, the RL agent's tuning of horizons and weights constrains QP size, and warm-starting further reduces solve time.

C. Priority-Based VRP Solution

The VRP component uses a greedy heuristic guided by RL priorities. Selecting the next node from *Na* candidates has complexity:

$$O(N_a^2)$$

which is polynomial and efficient for medium field sizes.

D. Overall Complexity

At deployment, the per-step complexity of RL–MPC is dominated by MPC optimization:

$$O\left(\left(N_a N_p\right)^3 + N_a^2\right),$$

With modest horizons $\left(N_p \leq 10\right)$and heuristic routing, the framework remains tractable for real-time operation. Empirically, our implementation achieves <1 ms per decision step for $N_a = 20$, $N_p = 5$, demonstrating scalability to practical field sizes.

VII. Conclusion

This paper introduced an RL–MPC framework for the pesticide VRP. In contrast to conventional approaches that optimize routing myopically at each step, the proposed method learns an adaptive control policy that jointly tunes pesticide allocation, routing priorities, and MPC horizons in response to real-time field conditions and forecasted pest dynamics.

In MATLAB simulations over a season-length horizon, RL–MPC achieved the lowest AURC, indicating superior overall pest suppression. While total travel and cumulative pesticide use were higher than GA–VRP and PSO–VRP, these increases reflect deliberate, risk-aware trade-offs: the policy favors timely, distributed interventions that prevent outbreaks and sustain mean field viability. The results underscore that long-term crop health cannot be achieved by minimizing travel or chemical use alone; effective policies must anticipate future pressures and act proactively. From a computational perspective, the trained RL policy provides constant-time inference (sub-millisecond), and the remaining per-step cost is dominated by a modest QP for MPC and a greedy, priority-weighted VRP heuristic—both scaling polynomially. End-to-end decision steps satisfied a 200 ms budget in our setup, supporting real-time use in autonomous operations. Overall, RL–MPC shifts the focus from short-term routing efficiency to long-horizon resilience in precision pest management. Future work will extend the framework to multivehicle coordination and depot logistics, incorporate richer stochastic weather/pest models and uncertainty-aware MPC, enforce environmental constraints (e.g., buffer zones, maximum per-area application), and validate performance in field trials at operational scale.

References

[1] H. Li, K. Liu, B. Yang, L. Zhang, and Y. Yan, "Path tracking of auwtonomous vehicle based on nmpc with pre-steering," IEEE Transactions on Consumer Electronics, vol. 70, pp. 966–979, 2024.

[2] R. Wang, J. Li, Q. Sun, H. Zhang, Z. Wei, and P. Wang, "Energy transfer converter between electric vehicles: Dc–dc converter based on virtual power model predictive control," IEEE Transactions on Consumer Electronics, vol. 69, no. 3, pp. 556–567, Aug. 2023.

[3] B. Wang, Z. Zha, L. Zhang, L. Liu, and H. Fan, "Deep reinforcement learning-based security-constrained battery scheduling in home energy system," IEEE Transactions on Consumer Electronics, vol. 70, no. 1, pp. 3548–3561, Feb. 2024.

[4] J. Wen, H. Dai, J. He, M. Xi et al., "Federated offline reinforcement learning with multimodal data," IEEE Transactions on Consumer Elec- tronics, vol. 70, no. 1, pp. 4266–4276, Feb. 2024.

[5] W. Cheng et al., "Blockchain and hybrid pso integration with fuzzy pid," IEEE Transactions on Consumer Electronics, vol. 70, no. 4, pp. 6965–6974, 2024.

[6] X. Lei et al., "Advanced algorithmic approaches for vrp variants in agricultural contexts," Journal/Conference Name, 2022, dOI or URL not provided.

[7] A. Weber, F. Fiege, and A. Knoll, "Vehicle mission guidance by symbolic optimal control," arXiv preprint, 2022.

[8] W. Kool, H. Van Hoof, and M. Welling, "Attention, learn to solve routing problems!" International Conference on Learning Representations (ICLR), 2019. [Online]. Available: https://openreview.net/forum?id=ByxBFsRqYm

[9] X. Fan, Z. Wang, and Y. Wang, "Rural business environments, informa- tion channels, and farmers' pesticide utilization behavior: a grounded theory analysis in hainan province, china," Agriculture, vol. 14, no. 2, p. 196, 2024.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Pro- ceedings of IEEE International Conference on Neural Networks, vol. 4. IEEE, 1995, pp. 1942–1948.

[11] Q. Hou, H. Zhang, L. Bao, Z. Song, C. Liu, Z. Jiang, and Z. Yang, "Ncs-delivered pesticides: a promising candidate in smart agriculture," International Journal of Molecular Sciences, vol. 22, no. 23, p. 13043, 2021.

[12] X. Kong, B. Zhang, and J. Wang, "Multiple roles of mesoporous silica in safe pesticide application by nanotechnology: a review," Journal of Agricultural and Food Chemistry, vol. 69, no. 24, pp. 6735–6754, 2021.

[13] L. Zhao, C. Wang, G. Hai-ying, and C. Yue, "Do chinese farmers misuse pesticide intentionally or not?" Agriculture, vol. 13, no. 9, p. 1749, 2023.

[14] J. B. Rawlings and D. Q. Mayne, Model Predictive Control: Theory and Design. Nob Hill Publishing, 2009.

[15] L. Gru¨ne and J. Pannek, Nonlinear Model Predictive Control: Theory and Algorithms, 2nd ed. Springer, 2017.

[16] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. MIT press, 2018.

[17] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 6, pp. 2042–2062, 2018.

[18] S. Gros, M. Zanon, and A. Bemporad, "Reinforcement learning for mixed-integer model predictive control," IEEE Transactions on Auto- matic Control, vol. 65, no. 11, pp. 4714–4726, 2020.

[19] D. Chaffey, W. Sun, and K. Sreenath, "Mpc-guided reinforcement learn- ing for

optimal control problems with unknown dynamics," in Learning for Dynamics and Control Conference. PMLR, 2022, pp. 403–415. [Online]. Available: https://proceedings.mlr.press/v168/chaffey22a.html

[20] M. Nazari, A. Oroojlooy, L. Snyder, and M. Taka´cˇ, "Reinforcement learning for solving the vehicle routing problem," in Advances in Neural Information Processing Systems (NeurIPS), 2018, pp. 9839–9849.