

Multi-Agent Deep Reinforcement Learning for Dynamic Depot Returns in Close Enough Multi-Vehicle Routing[★]

Francesca Guerriero^{a,*,1}, Francesco Paolo Saccomanno^b

ARTICLE INFO

Keywords:

Reinforcement Learning
Intelligent Systems
Precision Agriculture
Multiple Close Enough TSP
Matheuristics

ABSTRACT

The Multiple Close-Enough Traveling Salesman Problem (mCETSP) is a highly complex routing problem frequently encountered in the deployment of Unmanned Ground Vehicles (UGVs) for precision agriculture. This problem is formulated as a Mixed-Integer Non-Linear Programming (MINLP) problem of extreme complexity. To overcome this bottleneck, this paper proposes a novel hybrid framework that integrates Multi-Agent Deep Reinforcement Learning with exact mathematical optimization. By delegating the discrete return logic to the agents and the continuous spatial optimization to the mathematical solver, the proposed approach improves efficiency while avoiding the discretization errors typical of pure heuristic models. Extensive computational experiments demonstrate the efficiency of the proposed framework: the hybrid DRL-MINLP significantly limits computational times to about 20 seconds on all instances considered.

1. Introduction

The deployment of autonomous fleets, including Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs), has revolutionized numerous industrial domains, from advanced logistics to environmental monitoring and precision agriculture. In these operational scenarios, efficient path planning is crucial to minimize execution times, reduce energy consumption, and maximize overall mission yield. Traditionally, routing formulations are based on the classical Traveling Salesman Problem (TSP) or its multi-vehicle extensions (mTSP). However, in many real-world applications—such as pesticide spraying, RFID data harvesting, or visual inspections—a vehicle is not required to reach the exact center of a target. Instead, it is sufficient for the vehicle to enter a specific operational area, defined by a coverage radius or a neighborhood. This paradigm, formally known in the literature as the Close-Enough Traveling Salesman Problem (CE-TSP), offers significant operational advantages, allowing for a substantial reduction in total traveled distances compared to exact-point routing.

Transitioning from a single vehicle to a cooperative fleet introduces the Multiple Close-Enough Traveling Salesman Problem (mCETSP). While employing multiple vehicles in parallel effectively reduces the makespan (the completion time of the longest-running vehicle), it introduces non-trivial modeling challenges primarily linked to the physical and operational constraints of the machines. Moreover, UGVs and UAVs are bounded by finite autonomy, dictated by battery life or payload capacity (e.g., water or fertilizer tanks). Consequently, routing sequences must incorporate strategic and multiple returns to a central depot for recharging or resupplying. From a mathematical perspective, the mCETSP is formulated as a Mixed-Integer Non-Linear Programming (MINLP) problem of extreme complexity. The NP-hard nature of this problem stems from the coexistence of two highly coupled decision domains: a discrete domain (the assignment of targets to vehicles, i.e. the sequence of visits, and the insertion of depot returns) and a continuous domain (the determination of the exact geometric intersection coordinates within the target neighborhoods). In the literature, the majority of proposed approaches circumvent the complexity of the continuous domain by resorting to spatial discretization techniques. These methods sample a finite number of points within or on the perimeter of the targets, effectively reducing the problem to a Generalized TSP. Although this simplification allows for the application of traditional linear solvers, it inherently introduces discretization errors, precluding the discovery of the true global optimum and requiring a trade-off between the density of sampled points (which causes computation times to explode) and the quality of the solution. On the other hand, recent exact formulations in the continuous domain eliminate discretization errors entirely but suffer from severe computational intractability. As the number of targets increases,

[★]Corresponding author

✉ francesca.guerriero@unical.it (F. Guerriero); francescopaolo.saccomanno@unical.it (F.P. Saccomanno)
ORCID(s): <https://orcid.org/0000-0002-3887-1317> (F. Guerriero)

the combination of binary variables required to manage depot returns and continuous variables renders pure MINLP models inapplicable for realistic, large-scale instances. Concurrently, the use of Machine Learning, and particularly Reinforcement Learning (RL), has been explored to reduce the search space in the single-vehicle CE-TSP (for example, by pruning unlikely routing edges). However, it has not yet been extended to dynamically manage multi-agent logistics and recharging decisions within a continuous spatial framework.

To bridge this gap, this paper proposes a novel hybrid framework based on the integration of multi-agent Deep Reinforcement Learning and exact mathematical optimization. The proposed approach reduces the scalability limitations of pure MINLP models by delegating the resolution of the most critical discrete decision variables to a system of intelligent agents. Specifically, rather than utilizing RL to generate the entire routing sequence from scratch, the agents (one for each vehicle in the fleet) act as a high-level heuristic decision-maker. By observing the spatial distribution of upcoming targets and the baseline sequence, each agent learns a policy to dynamically shift the depot return points. The action of the agent dictates whether to anticipate, maintain, or delay the return to the base relative to the current sequence of visits. Then, by fixing the constraints related to depot returns through the output of the agent, the complexity of the residual problem is reduced. The underlying mathematical solver is then invoked to assign the target and to resolve the continuous spatial optimization, guaranteeing the total absence of discretization errors while minimizing the overall makespan.

The main contributions of this paper can be summarized as follows:

- A novel matheuristic architecture is formulated, hybridizing Multi-Agent Deep Reinforcement Learning with MINLP to solve the mCETSP.
- A decision-making paradigm is introduced wherein RL agents do not replace the exact solver but intelligently guide it by efficiently exploring the discrete space (dynamic determination of depot returns).
- Extensive computational experiments demonstrate that the proposed approach reduces resolution times compared to a pure MINLP model, while maintaining a level of geometric precision unattainable by discretization-based heuristics.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on the CE-TSP/mCETSP and ML-driven matheuristics. Section 3 defines the problem and presents the continuous mathematical model for the mCETSP. Section 4 details the proposed MADRL architecture, including the state space, action space, and reward function. Section 5 reports the computational results, benchmarking the hybrid approach against the pure MINLP model. Finally, Section 6 concludes the paper and outlines future research directions.

2. Literature Review

The problem addressed in this paper lies at the intersection of geometric routing optimization and artificial intelligence. Accordingly, the relevant literature can be categorized into two main streams: the algorithmic evolution of the CE-TSP and recent advancements in Machine Learning for Combinatorial Optimization (ML4CO).

Algorithmic Approaches to the CE-TSP The CE-TSP represents a significant shift from classical routing problems by relaxing the requirement of visiting exact nodes, allowing vehicles to intersect specific target neighborhoods instead. The primary challenge in solving the CE-TSP lies in its dual nature, requiring both the discrete optimization of the visiting sequence and the continuous geometric optimization of the intersection points, Mennell, Golden and Wasil (2011); Hà, Bostel, Langevin and Rousseau (2014); Behdani and Smith (2014). Historically, the most widely adopted resolution strategy has been spatial discretization, where the continuous neighborhood is transformed into a finite set of candidate waypoints on the perimeter, Carrabs, Cerrone, Cerulli and Gaudioso (2017). While this approach allows the deployment of well-established discrete solvers, it inherently suffers from discretization errors. To achieve high-quality solutions, the number of sampled points must be significantly increased, inevitably leading to an exponential growth in computational times and memory requirements. To circumvent these limitations, research has increasingly pivoted towards exact formulations in the continuous domain. Sophisticated models relying on branch-and-bound algorithms, Mixed-Integer Non-Linear Programming (MINLP), and Second-Order Cone Programming (SOCP) have been proposed to directly compute the exact continuous coordinates, Coutinho, Nascimento, Pessoa and Subramanian (2016); Di Placido, Archetti, Cerrone and Golden (2023); Zhang, Sauppe and Jacobson (2023); Lei and Hao (2024).

Recently, Araya, Angulo and Castro (2025) proposed exact decomposition schemes integrating polyhedral approximations of conic constraints to handle the slowness of standard SOCP solvers. Similarly, to address large-scale instances without relying on heavy SOCP solvers, Duong (2026) developed a quasi-linear heuristic exploiting hierarchical binary trees and fast local gradient descent, while Canturk and Dökeroğlu (2026) proposed scalable parallel memetic algorithms to maintain population diversity in complex geometric spaces.

When extending the continuous formulation to multi-vehicle fleets (mCETSP), the complexity further escalates. Carrabs, Cerulli, D'Ambrosio and Murano (2025) addressed the mCETSP for UAV fleets using a min-max routing objective to reduce the overall mission makespan. However, their exact resolution still relied on upper-bounding the problem through a discretized Multiple Generalized TSP (mGTSP) converted into MILP models. Building upon this, recent work by Guerriero and Saccomanno (2025b) formalized a pure continuous MINLP model for UGVs in precision agriculture that perfectly eliminates discretization errors and successfully handles autonomy constraints by scheduling depot returns. However, it highlights a critical bottleneck: combinatorial explosion. As the number of targets increases, the concurrent optimization of continuous geometric variables and binary sequence variables—particularly those governing dynamic returns to the depot—renders the pure MINLP model computationally prohibitive for realistic instances.

Machine Learning for Combinatorial Optimization and Matheuristics To overcome the scalability limits of exact mathematical programming, the operations research community has increasingly explored the integration of Machine Learning (ML), particularly Deep Reinforcement Learning (DRL), into combinatorial optimization. Initial efforts focused on end-to-end learning, where neural architectures generate routing sequences in a single pass. For instance, Fan, Cheng, Wu, Zhang, Yang, Wu and Sartoretti (2025) recently proposed UD3RL, a unified dual-decoder DRL framework for the CE-TSP that utilizes separate neural decoders for node sequencing and precise waypoint location, aided by k-nearest neighbors spatial reasoning. While these pure DRL models are highly efficient during inference, they typically struggle to strictly enforce hard operational constraints in densely constrained or dynamic scenarios, making them unsuitable for exact continuous geometric optimization with strict capacity limits. Consequently, recent state-of-the-art research has shifted from end-to-end solvers towards hybrid frameworks—often referred to as AI-driven matheuristics—where neural networks act as informed guides for exact solvers. These methods use ML for tasks such as branching, search space pruning, or discrete variable fixing. Notable examples include the 2LBB framework by You, Yang, Wang and Yin (2026), which utilizes a two-stage learning-to-branch approach to identify promising branching candidates in exact Vehicle Routing Problem (VRP) solvers, and the Apollo-MILP framework by Liu, Wang, Geng, Li, Zong, Zhu, Hao and Wu (2025). Apollo-MILP specifically tackles the risk of infeasibility (often called "neural diving") by introducing an alternating prediction-correction cycle where neural predictions are corrected by exact solver trust-region searches. Despite these remarkable advancements in AI-OR hybridization, the application of such frameworks to geometric routing problems remains extremely scarce. A preliminary step was taken by Guerriero and Saccomanno (2025a), who employed an ML agent to predict and prune unlikely edges in a single-vehicle CE-TSP, shrinking the feasible space before applying a mathematical solver.

However, to the best of the authors' knowledge, no existing literature has successfully extended this hybrid paradigm to manage the multi-agent logistics of the mCETSP. Specifically, the dynamic shifting of depot returns under capacity constraints—a purely discrete decision that heavily dictates the feasibility of the continuous spatial optimization—has never been formulated as an RL-guided matheuristic.

The present work fills this crucial gap. By deploying a Multi-Agent Deep Reinforcement Learning architecture to control the discrete shifting of depot return points, this paper proposes a novel framework where the AI intelligently navigates the combinatorial complexity, empowering the underlying exact MINLP solver to efficiently and optimally resolve the continuous geometric coordinates without scalability issues.

3. Problem Definition and Mathematical Formulation

For completeness and to improve readability, in this section we briefly restate the mCETSP mathematical formulation previously introduced in Guerriero and Saccomanno (2025b). The model adopts a sequence-based approach rather than a classical flow-based graph formulation, a structural choice designed to isolate the discrete routing decisions from the continuous geometric optimization.

The problem is defined on a two-dimensional Euclidean plane, given the area boundaries M_x and M_y , containing a central depot with coordinates (x_0^t, y_0^t) and a set of N target points. Each target $j \in \{1, \dots, N\}$ is located at coordinates

(x_j^t, y_j^t) and is associated with a continuous circular neighborhood defined by a coverage radius ρ_j . A fleet of R homogeneous UGVs is stationed at the depot.

The routing is modeled through a predefined sequence of N “Operation Points” (OPs). The fleet must collectively visit all targets such that each target is covered by exactly one OP. The objective is to minimize the makespan $C_{\max}$, defined as the longest route traveled by any single UGV in the fleet, while allowing strategic returns to the depot to satisfy the vehicles’ operational constraints.

Let (x_k, y_k) be the continuous coordinate variables of the k -th OP. The distance between consecutive OPs is denoted by d_k , while c_k tracks the cumulative distance traveled up to OP k . The assignment of target j to OP k is handled by the binary variable $\alpha_{kj} \in \{0, 1\}$. The crucial logistical decision is governed by the route-splitting binary variable $r_k \in \{0, 1\}$, which equals 1 if the UGV returns to the depot, and 0 otherwise.

The problem is formulated as the following MINLP:

$$\min \quad C_{\max} \quad (1a)$$

$$\text{s.t.} \quad c_1 = d_1 + (1 - r_1)d_0 \quad (1b)$$

$$c_k = d_k + (1 - r_k)c_{k-1} \quad \forall k \in \{2, \dots, N-1\} \quad (1c)$$

$$c_N = c_{N-1} + d_N \quad (1d)$$

$$c_k \leq C_{\max} \quad \forall k \in \{1, \dots, N\} \quad (1e)$$

$$\sum_{k=1}^{N-1} r_k = R - 1 \quad (1f)$$

$$d_k^2 \geq w_k^2 + u_k^2 \quad \forall k \in \{0, \dots, N\} \quad (1g)$$

$$w_k = x_{k+1} - x_k \quad \forall k \in \{1, \dots, N-1\} \quad (1h)$$

$$w_0 = x_1 - x_0^t \quad (1i)$$

$$w_N = x_N - x_0^t \quad (1j)$$

$$u_k = y_{k+1} - y_k \quad \forall k \in \{1, \dots, N-1\} \quad (1k)$$

$$u_0 = y_1 - y_0^t \quad (1l)$$

$$u_N = y_N - y_0^t \quad (1m)$$

$$\sum_{j=1}^N \alpha_{kj} = 1 \quad \forall k \in \{1, \dots, N\} \quad (1n)$$

$$\sum_{k=1}^N \alpha_{kj} = 1 \quad \forall j \in \{1, \dots, N\} \quad (1o)$$

$$t_k^x = x_k - \sum_{j=1}^N \alpha_{kj} x_j^t \quad \forall k \in \{1, \dots, N\} \quad (1p)$$

$$t_k^y = y_k - \sum_{j=1}^N \alpha_{kj} y_j^t \quad \forall k \in \{1, \dots, N\} \quad (1q)$$

$$(t_k^x)^2 + (t_k^y)^2 \leq (\rho_k)^2 \quad \forall k \in \{1, \dots, N\} \quad (1r)$$

$$|x_k - x_0^t| \leq M_x(1 - r_k) \quad \forall k \in \{1, \dots, N-1\} \quad (1s)$$

$$|y_k - y_0^t| \leq M_y(1 - r_k) \quad \forall k \in \{1, \dots, N-1\} \quad (1t)$$

$$r_k, \alpha_{kj} \in \{0, 1\}, \quad C_{\max}, c_k, d_k \geq 0 \quad (1u)$$

The objective function (1a) minimizes the makespan, enforcing simultaneous task completion. Constraints (1b)–(1d) evaluate the cumulative distance c_k along the sequence. Crucially, the formulation exploits the variable r_k : if a return to the depot is triggered ($r_k = 1$), the term $(1 - r_k)$ zeros out the accumulated distance, effectively resetting the c_k counter for the subsequent UGV. Constraint (1e) bounds these cumulative route segments by the global makespan.

Constraint (1f) implicitly manages the multi-vehicle fleet by enforcing exactly $R - 1$ route splits, effectively dividing the sequence of N operations into R distinct sub-tours. Constraints (1g)–(1m) define the Euclidean distances between consecutive OPs through auxiliary coordinate differences w_k and u_k . The proper assignment between OPs and physical targets is guaranteed by (1n) and (1o). The core CE-TSP geometry is captured by (1p)–(1r), ensuring that the exact optimized coordinates (x_k, y_k) fall within the circular radius ρ_k of the assigned target. Finally, Constraints (1s) and (1t) utilize the area boundaries M_x and M_y to enforce the physical return: if $r_k = 1$, the OP is mathematically forced to collapse onto the exact coordinates of the central depot.

The sequence-based formulation accurately solves the continuous mCETSP without discretization errors. However, as the number of targets increases, the concurrent optimization of the assignment matrix (α_{kj}) , the continuous intersection coordinates $((x_k, y_k))$, and the route-splitting variables (r_k) leads to severe computational intractability Guerriero and Saccomanno (2025b). Specifically, the solver struggles with the combinatorial explosion of deciding exactly *where* to split the route. Evaluating every possible combination of r_k to manage the autonomy of the UGV forces the solver to explore an exponentially growing branch-and-bound tree. This indicates that, to scale the model to realistic agricultural instances, the discrete logic should be predicted and fixed externally, leaving the solver to rapidly resolve only the continuous spatial coordinates.

This critical need motivates the introduction of the hybrid Multi-Agent Reinforcement Learning architecture detailed in the following section.

4. The Hybrid DRL-MINLP Matheuristic Framework

In order to reduce the combinatorial complexity of the model, we proposed a hybrid matheuristic framework that delegates the discrete route-splitting decisions to an agent, based on Deep Reinforcement Learning (DRL), leaving the MINLP solver to optimize only the continuous coordinates (x_k, y_k) and the assignment matrix α_{kj} . This architecture ensures that the neural network exclusively manages the discrete logistics of depot returns, while the mathematical programming framework guarantees strict adherence to the complex continuous geometric boundaries of the mCETSP.

The core intuition of our approach is based on *iterative shifting*. Consider the base case of $R = 2$ vehicles, which requires exactly one depot return (one split in the sequence). Initially, this return point is assigned randomly at some OP index k . The agent is trained to evaluate this configuration and decide whether the return should occur *earlier* or *later* in the sequence, and by how many steps. This logic seamlessly scales in the case of a greater number of vehicles ($R > 2$): the same trained policy evaluates multiple initial random return points simultaneously, deciding the optimal shift for each one at every execution step.

MDP Formulation for Dynamic Sequence Shifting To train the DRL agent, we adopt the Proximal Policy Optimization (PPO) algorithm, a state-of-the-art policy gradient method chosen for its stability and its ability to handle continuous action spaces. Then, the discrete decision of fixing the r_k variables is modeled as a Markov Decision Process (MDP) defined by the tuple $(S, \mathcal{A}, \mathcal{P}, \mathcal{R})$.

State Space (S): At any step t , the state s_t provides the agent with the necessary spatial and logistical context. It encapsulates the current configuration of the operation sequence, the relative distances along the sequence, and the indices of the $R-1$ depot returns. Specifically, it is represented as a sequence of feature vectors for each OP k , including: (i) the Euclidean distance from the depot to OP k , (ii) the distance from OP k to OP $k+1$, and (iii) a positional encoding indicating the current locations of the route splits (i.e., the indices where $r_k = 1$). By observing this context, the agent learns to identify distance bottlenecks and workload imbalances among the induced sub-tours, determining whether a split should be shifted to better distribute the travel distance across the fleet.

Action Space ($\mathcal{A}$): The most critical aspect of the framework is the action design. Let $K_{split}^{(t)} = \{k_1, k_2, \dots, k_{R-1}\}$ be the set of indices where a depot return is currently scheduled ($r_k = 1$). For each split point $k_i \in K_{split}^{(t)}$, the DRL agent outputs a continuous action $a_i \in [-1, 1]$. If $a_i < 0$, the vehicle returns to the depot earlier; if $a_i > 0$, it returns later. To map this continuous output to a discrete sequence shift, we introduce a scaling factor Ω , which defines the maximum allowed shift window (i.e., the maximum number of nodes the split can be moved in a single step). The discrete shift δ_i is computed as:

$$\delta_i = \lfloor a_i \times \Omega \rfloor \quad (2)$$

where $\lfloor \cdot \rfloor$ denotes the nearest integer rounding. The updated split position for the next step becomes $k_i^{(t+1)} = k_i^{(t)} + \delta_i$.

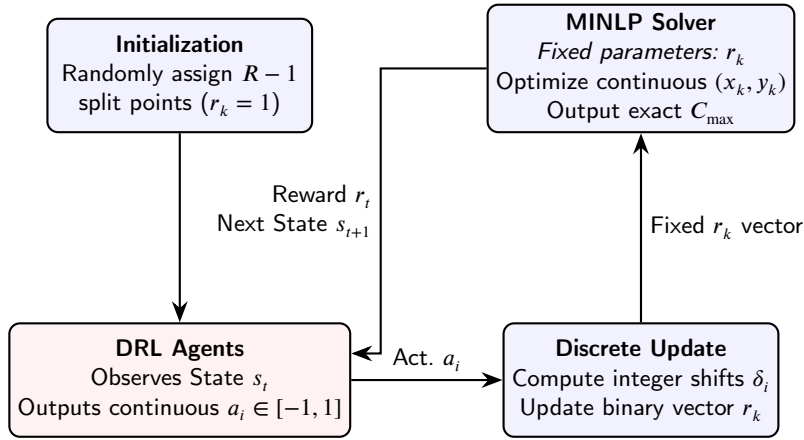


Figure 1: The proposed DRL-MINLP Matheuristic Execution Cycle. The agent handles the combinatorial shifting of depot returns (discrete logic), while the MINLP solver evaluates the exact routing geometry (continuous optimization).

Reward Function ($\mathcal{R}$): The learning signal r_t must reflect the quality of the shifts. Once the agent updates the split points, the new binary vector r_k is fixed and passed to the MINLP solver. The solver, now with reduced complexity, computes the continuous coordinates and returns the objective value $C_{\max}$ (the fleet's makespan). The reward is formulated to penalize configurations that yield a higher $C_{\max}$, pushing the agent to find the shifts that balance the workload across the fleet while minimizing travel distance. For the considered objective, the reward function is formulated to guide the agent directly toward the minimization of the fleet's makespan $C_{\max}$. To provide an informative learning signal at each step, we adopt a differential reward structure. Let $C_{\max}^{(t-1)}$ be the makespan computed by the exact MINLP solver prior to the agent's action, and $C_{\max}^{(t)}$ be the newly computed makespan after the sequence shifts δ_i have been applied. The reward at step t is defined as $r_t = C_{\max}^{(t-1)} - C_{\max}^{(t)}$. Under this formulation, the agent receives a positive reward if the proposed shifts reduce the longest route, and a negative penalty otherwise.

The Matheuristic Execution Cycle We can summarize the flow of the framework in the following steps (Fig. 1):

1. **Initialization:** Given a sequence of N targets and R vehicles, $R - 1$ variables r_k are randomly set to 1, while the rest are set to 0.
2. **Discrete optimization:** The agent observes the current state and generates the continuous actions a_i . The shifts δ_i are computed, and the r_k vector is updated accordingly, moving the depot returns backward or forward in the sequence.
3. **Exact Solving:** The updated r_k variables are treated as hard constants (fixed parameters) within the model (1) and the MINLP solver optimizes the continuous neighborhood intersections (x_k, y_k) and assignments of targets to OPs.
4. **Feedback:** The computed $C_{\max}$ is used to evaluate the effectiveness of the shift, and the cycle repeats or terminates based on a predefined stopping criterion (e.g., maximum iterations or convergence of the makespan).

5. Computational Experiments

To validate the effectiveness and scalability of the proposed DRL-MINLP matheuristic framework, an extensive series of computational experiments was conducted. The primary objective of these experiments is to demonstrate how the integration of the agent accelerates the resolution of the continuous mCETSP compared to a pure mathematical approach, particularly as the number of targets increases.

All training phases and computational evaluations were executed on a workstation equipped with a Windows 11 operating system, an Intel Core i9 processor (2.20 GHz), and 64 GB of RAM. The framework was implemented in Python. To seamlessly integrate the mathematical optimization with the reinforcement learning architecture, the environment was custom-built leveraging the Gymnasium API. This standardized interface allowed the environment

Table 1

Hyperparameters configuration for the PPO agent.

Hyperparameter	Value
Learning Rate (α)	3×10^{-4}
Discount Factor (γ)	0.99
GAE Parameter (λ)	0.95
Clipping Range (ϵ)	0.2
Value Function Coefficient	0.5
Entropy Coefficient	0.0
Batch Size	64
Number of Steps (n_steps)	2048
Network Architecture	MLP ([128, 128])

to efficiently communicate states, actions, and rewards to the learning agent. The PPO algorithm was implemented utilizing the `Stable-Baselines3` library, a highly reliable and open-source framework based on PyTorch. The optimization was delegated to the Gurobi 12.0.1 solver.

The test instances were generated extending the widely adopted benchmark dataset proposed by Behdani et al (Behdani and Smith (2014)). The coordinates for both the central depot and the targets were randomly distributed within a 16×10 unit area. To comprehensively evaluate the impact of the continuous neighborhoods on the routing sequences, experiments were conducted using three different coverage radii: $\rho \in \{0.25, 0.50, 1.00\}$. Crucially, while previous pure MINLP evaluations were limited to small instances (see Guerriero and Saccomanno (2025b)), the hybrid framework was tested on configurations ranging from 6 up to 20 target nodes. The neural network architecture for the DRL agent consists of a Multi-Layer Perceptron (MLP) feature extractor. The core hyperparameters governing the learning process are summarized in Table 1. These parameters were selected to ensure stable policy updates and a balanced exploration-exploitation trade-off within the continuous action space.

To rigorously assess the computational advantages of the proposed hybrid matheuristic, its performance was benchmarked against the exact pure MINLP model C-mCETSP (Guerriero and Saccomanno (2025b)) solved natively by Gurobi. The comparison is evaluated across two primary metrics: the objective value ($C_{\max}$) and the total computational time. Tables 2 and 3 summarize the comparative results for varying problem sizes ($N \in \{6, 8, 10\}$) and coverage radii ($\rho \in \{0.25, 0.50, 1.00\}$), considering scenarios with 2 and 3 vehicles. The columns $C_{\max}$ and Time report the makespan and execution time for the C-mCETSP and DRL-MINLP methods, respectively. The final column shows the percentage difference δ between the two approaches. It quantifies the relative degradation in the makespan achieved by the DRL-MINLP approach with respect to the exact solution computed by C-mCETSP. It is calculated as follows:

$$\delta = \left(\frac{C_{\max}^{\text{DRL-MINLP}} - C_{\max}^{\text{C-mCETSP}}}{C_{\max}^{\text{DRL-MINLP}}} \right) \times 100, \quad (3)$$

where $C_{\max}^{\text{C-mCETSP}}$ and $C_{\max}^{\text{DRL-MINLP}}$ represent the maximum completion times (makespan) obtained by the exact benchmark and the proposed deep reinforcement learning method, respectively. A lower value of δ indicates that the learning-based framework yields a solution closer to the exact optimum.

The computational results clearly highlight the practical relevance of the problem across multiple application domains. In particular, the results indicate that optimizing routes by exploiting target neighborhoods, rather than enforcing visits to exact target points, leads to more efficient solutions. Moreover, the results confirm that increasing the coverage radius (ρ) consistently reduces the global makespan, demonstrating the solver's ability to leverage continuous neighborhoods effectively once the discrete logistical decisions are removed (or substantially reduced).

Conversely, the empirical findings clearly highlight the theoretical bottleneck discussed in Section 3. For small instances ($N = 6$), the pure MINLP solver finds the optimal solution in a few seconds, with runtimes comparable to the DRL-MINLP approach. However, when the problem size increases to $N = 10$, the combinatorial growth of the binary variables associated with depot returns drives the exact solver's runtime up to 888 seconds in the worst case.

In contrast, the hybrid DRL-MINLP framework reduces this combinatorial barrier. By relying on the agent to predict high-quality split points, the Gurobi solver is tasked with a lower complexity and the exact continuous geometric

Table 2

Comparison of C-mCETSP and DRL-MINLP for two vehicles.

Instance		C-mCETSP		DRL-MINLP		
N	ρ	$C_{\max}$	Time (s)	$C_{\max}$	Time (s)	δ (%)
6	0.25	22.35	0.27	23.29	0.47	4.03
	0.50	21.58	0.31	22.49	0.53	4.04
	1.00	20.21	0.37	20.83	0.61	2.98
8	0.25	25.02	2.90	25.81	1.39	3.06
	0.50	24.22	3.46	25.35	1.59	4.46
	1.00	22.72	7.29	23.57	2.59	3.61
10	0.25	26.23	29.37	26.70	8.75	1.76
	0.50	25.28	57.15	25.75	9.98	1.82
	1.00	23.63	91.23	23.86	20.86	0.96

Table 3

Comparison of C-mCETSP and DRL-MINLP for three vehicles.

Instance		C-mCETSP		DRL-MINLP		
N	ρ	$C_{\max}$	Time (s)	$C_{\max}$	Time (s)	δ (%)
6	0.25	21.03	0.81	22.83	0.49	7.89
	0.50	20.48	1.01	21.74	0.55	5.80
	1.00	19.37	1.95	20.18	0.63	4.01
8	0.25	23.19	6.12	23.58	1.20	1.65
	0.50	22.62	7.25	23.16	1.43	2.33
	1.00	21.51	13.69	21.87	2.34	1.64
10	0.25	24.16	296.69	24.42	7.63	1.06
	0.50	23.53	570.15	23.74	9.23	0.88
	1.00	22.31	888.15	22.37	16.08	0.27

optimization. As a result, the hybrid approach identifies near-optimal solutions (with a gap generally below 2% for larger instances) within 20 seconds in the worst case.

The superiority of the hybrid approach becomes more evident in the three-vehicle setting. In this case, DRL-MINLP scales better because it more substantially reduces the number of constraints by fixing the depot-return decisions for all three vehicles. As a result, it achieves consistently shorter runtimes (always below 17 seconds), whereas the pure mathematical model exceeds 800 seconds in the worst case, in this most challenging configuration. In terms of solution quality, the gap remains limited: in the most challenging instances, the relative deviation stays below 1%.

6. Conclusions and Future Work

This paper presented a novel hybrid matheuristic framework designed to solve the continuous mCETSP applied to UGVs in precision agriculture. The core methodological contribution of this work lies in the explicit decoupling of discrete logistical decisions from continuous geometric optimization. By conceptualizing the mCETSP as a sequence-based routing problem, we successfully delegated the intractable combinatorial logic—specifically the dynamic insertion of intermediate depot returns to satisfy operational autonomy—to a DRL agent trained via PPO.

The empirical results demonstrate the computational advantages of this hybrid architecture. While the exact pure MINLP formulation suffers from exponential combinatorial explosion, failing to converge or running out of memory for instances with more than 15 targets, the proposed DRL-MINLP framework scales to 10 targets. By predicting accurate discrete route-splitting configurations, the neural network clears the path for the MINLP solver, which computes the exact continuous intersection coordinates within the target neighborhoods in a matter of seconds, maintaining an

optimality gap near 1% for the most difficult cases. In conclusion, the proposed matheuristic confirmed that learning-based spatial shifting can break the computational bottlenecks of continuous routing models, paving the way for scalable fleet management in modern precision agriculture.

Although the proposed framework is robust and theoretically sound, this foundational work opens several promising avenues for future research. First, the current spatial environment assumes a deterministic, obstacle-free two-dimensional Euclidean plane. Future iterations of the model could seamlessly integrate obstacle-avoidance constraints and multi-elevation 3D topographical data to better reflect complex agricultural terrains. Second, while the PPO agent currently relies on a standard MLP leveraging handcrafted geometric features, representing the state space as a graph could allow the deployment of Graph Neural Networks, potentially enhancing the agent's ability to capture complex topological symmetries in larger instances. Finally, the present model assumes deterministic vehicle parameters. Extending the framework to account for stochastic features, such as battery depletion rates, driven by payload variations or dynamic weather conditions, would further bridge the gap between mathematical optimization and real-world robotic deployment.

References

- Araya, D., Angulo, G., Castro, M.P., 2025. Exact and approximate formulations for the close-enough TSP. URL: <https://optimization-online.org/?p=32840>.
- Behdani, B., Smith, J.C., 2014. An Integer-Programming-Based Approach to the Close-Enough Traveling Salesman Problem. *INFORMS Journal on Computing* 26, 415–432. URL: <https://pubsonline.informs.org/doi/10.1287/ijoc.2013.0574>, doi:10.1287/ijoc.2013.0574.
- Canturk, D., Dökeoğlu, T., 2026. Scalable parallel memetic algorithms for large close-enough traveling salesman problems. *Neural Computing and Applications* 38, 27. URL: <https://doi.org/10.1007/s00521-025-11706-4>, doi:10.1007/s00521-025-11706-4.
- Carrabs, F., Cerrone, C., Cerulli, R., Gaudioso, M., 2017. A novel discretization scheme for the close enough traveling salesman problem. *Computers & Operations Research* 78, 163–171. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0305054816302179>, doi:10.1016/j.cor.2016.09.003.
- Carrabs, F., Cerulli, R., D'Ambrosio, C., Murano, G., 2025. Upper Bound Computation for the Multiple Close-Enough Traveling Salesman Problem., in: *Proceedings of the 14th International Conference on Operations Research and Enterprise Systems, SCITEPRESS - Science and Technology Publications, Porto, Portugal*. pp. 186–195. URL: <https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0013377900003893>, doi:10.5220/0013377900003893.
- Coutinho, W.P., Nascimento, R.Q.D., Pessoa, A.A., Subramanian, A., 2016. A Branch-and-Bound Algorithm for the Close-Enough Traveling Salesman Problem. *INFORMS Journal on Computing* 28, 752–765. URL: <https://pubsonline.informs.org/doi/10.1287/ijoc.2016.0711>, doi:10.1287/ijoc.2016.0711.
- Di Placido, A., Archetti, C., Cerrone, C., Golden, B., 2023. The generalized close enough traveling salesman problem. *European Journal of Operational Research* 310, 974–991. URL: <https://linkinghub.elsevier.com/retrieve/pii/S037722172300293X>, doi:10.1016/j.ejor.2023.04.010.
- Duong, K., 2026. A Fast Quasi-Linear Heuristic for the Close-Enough Traveling Salesman Problem. URL: <https://arxiv.org/abs/2603.21401>, doi:10.48550/ARXIV.2603.21401.
- Fan, M., Cheng, J., Wu, Y., Zhang, Y., Yang, Y., Wu, G., Sartoretti, G., 2025. A Unified Deep Reinforcement Learning Approach for Close Enough Traveling Salesman Problem. URL: <http://arxiv.org/abs/2510.03065>, doi:10.48550/arXiv.2510.03065, arXiv:2510.03065.
- Guerriero, F., Saccomanno, F., 2025a. A Machine Learning Matheuristic for Agricultural Drone Routing, in: *Proceedings of the 13th International Symposium on Intelligent Manufacturing and Service Systems, IMSS*. pp. 322–330. URL: <https://doi.org/10.5281/zenodo.17530789>, doi:10.5281/zenodo.17530789.
- Guerriero, F., Saccomanno, F.P., 2025b. Optimal routing of unmanned ground vehicles using the multiple close-enough traveling salesman problem, in: *Proceedings of the IEEE International Congress on Smart Agriculture and Sustainable Systems Engineering, IEEE, Marrakech, Morocco*.
- Hà, M.H., Bostel, N., Langevin, A., Rousseau, L., 2014. Solving the close-enough arc routing problem. *Networks* 63, 107–118. URL: <https://onlinelibrary.wiley.com/doi/10.1002/net.21525>, doi:10.1002/net.21525.
- Lei, Z., Hao, J.K., 2024. An effective memetic algorithm for the close-enough traveling salesman problem. *Applied Soft Computing* 153, 111266. URL: <https://linkinghub.elsevier.com/retrieve/pii/S1568494624000401>, doi:10.1016/j.asoc.2024.111266.
- Liu, H., Wang, J., Geng, Z., Li, X., Zong, Y., Zhu, F., Hao, J., Wu, F., 2025. Apollo-MILP: An Alternating Prediction-Correction Neural Solving Framework for Mixed-Integer Linear Programming. URL: <http://arxiv.org/abs/2503.01129>, doi:10.48550/arXiv.2503.01129, arXiv:2503.01129.
- Mennell, W.K., Golden, B.L., Wasil, E., 2011. A Steiner-Zone Heuristic for Solving the Close-Enough Traveling Salesman Problem, in: *12th INFORMS Computing Society Conference, INFORMS*. pp. 162–184. URL: <http://www.informs.org/Community/Conferences/ICS2011/Conference-Volume>, doi:10.1287/ics.2011.0004.
- You, Z., Yang, Y., Wang, X., Yin, W., 2026. Two-Stage Learning to Branch in Branch-Price-and-Cut Algorithms for Solving Vehicle Routing Problems Exactly. *Operations Research*, opre.2023.0615 URL: <https://pubsonline.informs.org/doi/10.1287/opre.2023.0615>, doi:10.1287/opre.2023.0615.
- Zhang, W., Sauppe, J.J., Jacobson, S.H., 2023. Results for the close-enough traveling salesman problem with a branch-and-bound algorithm. *Computational Optimization and Applications* 85, 369–407. URL: <https://link.springer.com/10.1007/s10589-023-00474-3>, doi:10.1007/s10589-023-00474-3.