Reinforcement Learning Based Collaborative Path Planning Research for UAVs and Unmanned Vehicles

Xuanran Li LIXUANRAN@GS.ZZU.EDU.CN
School of Cyber Science and Engineering, Zhengzhou University
Longxin Yao YLX15138699692@GS.ZZU.EDU.CN
School of Cyber Science and Engineering, Zhengzhou University
Mingzhe Li 1351147614@QQ.COM
School of Cyber Science and Engineering, Zhengzhou University
Bo Zhang* ZHANGBO2050@ZZU.EDU.CN
School of Cyber Science and Engineering, Zhengzhou University

Editors: Nianyin Zeng, Ram Bilas Pachori and Dongshu Wang

Abstract

This paper presents a reinforcement learning framework for multi-UAV and UGV coordinated path planning with charging constraints. We formulate the problem as a Markov Decision Process and develop a Transformer-based solution combining encoder-decoder architecture with policy gradients to optimize path synchronization and charging coordination. Experimental results demonstrate that our approach outperforms existing heuristic methods (GLS, TS) in terms of solution quality and generalization across different problem scales. The proposed method effectively minimizes mission completion time while handling energy constraints through intelligent charging point synchronization.

Keywords: UAV-UGV Collaboration; Path Planning; Reinforcement Learning; Energy-Constrained UAVs

1. Introduction

Unmanned Aerial Vehicles (UAVs) have emerged as a transformative technology with extensive military and civilian applications, including but not limited to traffic surveillance, border security, disaster management, and forest fire monitoring (Kandris et al., 2020). These applications typically require sustained UAV operation, which is fundamentally constrained by limited battery capacity. To extend mission endurance and enhance operational efficiency, Unmanned Ground Vehicles (UGVs) can be employed as mobile charging stations to provide in-situ power replenishment for UAVs (Paudel et al., 2024a). This multi-agent cooperative path planning problem necessitates not only the optimization of individual trajectories for both UAVs and UGVs, but also the precise synchronization of charging rendezvous to ensure mission completion with maximal efficiency (Cai et al., 2024).

In this study, we investigate a system comprising multiple energy-constrained UAVs and a single UGV, tasked with visiting a set of predetermined mission points while minimizing total operation time. The limited endurance of each UAV mandates periodic recharging from the UGV (functioning as a mobile charging platform), thereby transforming the path planning challenge into a coupled optimization problem that must simultaneously address: (1) optimal trajectory generation for all agents, and (2) temporal-spatial coordination of charging events to maintain uninterrupted mission execution.

2. Related work

Recent research on UAV-UGV cooperative systems has explored various approaches to path planning and energy management. Traditional optimization methods, including exact algorithms (Tilk et al., 2018) and metaheuristics (Nonut et al., 2022; Zhang, 2024), provide theoretical foundations but struggle with scalability and dynamic adaptation. Learning-based techniques such as Graph Neural Networks (Li et al., 2024) and Deep Reinforcement Learning (Zhou et al.) have shown promise in handling complex coordination tasks, though they often overlook critical charging constraints in their formulations.

Energy management solutions have evolved along three main directions. Static charging systems (Paudel et al., 2024b) offer simplicity but limit operational flexibility. Mobile charging approaches (Huang et al., 2025a) provide greater adaptability at the cost of increased computational complexity. Hybrid systems (Huang et al., 2025b) attempt to balance these trade-offs but introduce new challenges in reliability and implementation.

Despite these advances, current approaches share several limitations (Mondal et al., 2024). Most methods treat path planning and charging coordination as separate problems, leading to suboptimal system performance. Existing solutions also demonstrate limited scalability when applied to larger problem instances. Furthermore, the dynamic nature of real-world environments poses significant challenges that are not adequately addressed by current frameworks. Our work aims to overcome these limitations through an integrated approach that jointly optimizes both path planning and charging coordination.

We present a novel Transformer-based reinforcement learning framework for UAV-UGV cooperative path planning with three key contributions: First, we establish a Markov Decision Process formulation that jointly optimizes trajectory planning and charging coordination through Transformer architectures. Second, we demonstrate the framework's strong generalization capability across varying problem scales and spatial distributions. Third, extensive experiments show our method achieves superior performance compared to state-of-the-art heuristic approaches in both solution quality and computational efficiency.

3. Formulation of the problem

3.1. Description of the problem

To formally define the problem, let us consider a system consisting of n fuel-constrained Unmanned Aerial Vehicles (UAVs), $i \in \{1, 2, \ldots, n\}$, an Unmanned Vehicle (UGV), and k mission points to be visited. There are two types of task points: one is located in the road network G and can be accessed by a UAV hovering or a UGV road; the other access points are located outside the road network and are only accessed by a UAV. A task point can be defined as the set $\{1, 2, ..., k\}$. UGV and UAVs exhibit different characteristics: while UAVs have limited battery capacity but can fly at faster speeds, UGV move at slower speeds and operate only on the road network G. The UAVs and UGV have different characteristics. The goal is to minimize the total time required to complete all mission point visits while ensuring that each UAV has sufficient fuel levels to complete the mission and be able to reach the UAV for recharging if needed. The challenges are therefore manifold, including rationalizing the assignment of mission points to each UAV to avoid duplicate mission point visits and conflicts; planning optimal paths for each UAV and UGV so that they can efficiently visit the mission points and recharge; and coordinating the UAV's recharging needs to ensure that the UAV is able to recharge each UAV at the right time and place.

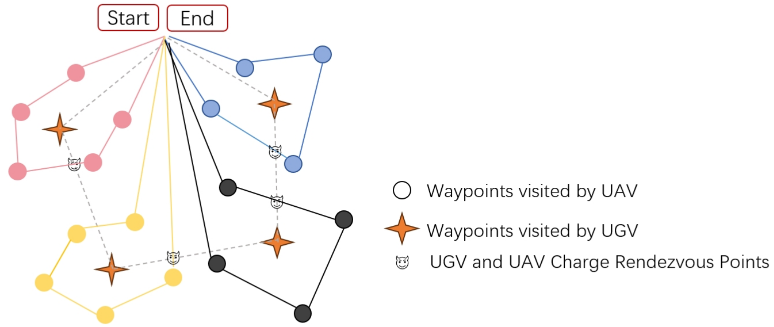


Figure 1: Schematic of the fuel-constrained UAV-UGV co-routing problem

The charging coordination system activates when a UAV's energy level drops below a predefined threshold, triggering an automatic charging request to the nearest available UGV. The system employs an optimized path planning algorithm for the UGV that considers multiple factors including remaining mission requirements and relative distances between agents. A fixed charging duration is incorporated into the planning framework to account for temporal constraints. This approach ensures efficient energy replenishment while maintaining mission continuity through dynamic adjustment of charging schedules based on real-time system states.

3.2. MDP process

The problem can be modeled as a sequential decision-making system as shown in the figure2 below.

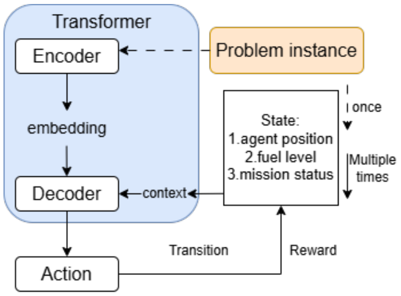


Figure 2: MDP representation for the UAV-UGV cooperative routing problem utilizing a Transformer network

The system can be represented as a Markov Decision Process (MDP) with components defined as tuples $< S, A, R, T >$ as follows:

1) **The state space (S)**: The environment state s is defined as $s = (p_{\text{UAV}}, f_{\text{UAV}}, v_{\text{UAV}}, p_{\text{UGV}}, v_{\text{UGV}}, v_{\text{task}})$, where $p_{\text{UAV}} = [p_{\text{UAV},1}, p_{\text{UAV},2}, \ldots, p_{\text{UAV},n}]$ represents the positions of each UAV, $f_{\text{UAV}} = [f_{\text{UAV},1}, f_{\text{UAV},2}, \ldots, f_{\text{UAV},n}]$ denotes the fuel levels of each UAV, $v_{\text{UAV}} = [v_{\text{UAV},1}, v_{\text{UAV},2}, \ldots, v_{\text{UAV},n}]$

indicates the list of task points visited by each UAV, p_{UGV} is the position of the UGV, v_{UGV} represents the list of task points visited by the UGV, and v_{task} is a binary vector indicating the visit status of each task point (1 for visited).

2)**The action space (A)**: comprises all possible actions of UAVs and the UGV, where an action a is defined as $a = (a_{\text{UAV}}, a_{\text{UGV}})$. Here, $a_{\text{UAV}} = [a_{\text{UAV},1}, a_{\text{UAV},2}, \ldots, a_{\text{UAV},n}]$ represents the actions of each UAV, which can include visiting a specific task point or flying to the UGV for charging, and a_{UGV} denotes the action of the UGV, which may involve moving to a task point or a charging station.

3)**The reward function (R)**: The reward function is defined as $R(s,a) = -(\text{Time}_{\text{UAV}} + \text{Time}_{\text{UGV}} + P_{\text{fail}} \cdot I_{\text{fail}})$, where Time_{UAV} is the total flight time of UAVs, Time_{UGV} is the travel time of the UGV, P_{fail} is the penalty for task failure, and I_{fail} is an indicator function that equals 1 if the task fails and 0 otherwise.

4)**The transition function (T)**: The next state s' is defined as $s' = (P'_{\text{UAV}}, f'_{\text{UAV}}, v'_{\text{UAV}}, P'_{\text{UGV}}, v'_{\text{UGV}}, v'_{\text{task}})$, where P'_{UAV}, f'_{UAV}, and v'_{UAV} represent the updated positions, fuel levels, and task point visitation statuses of the UAVs, respectively. Similarly, P'_{UGV} and v'_{UGV} denote the updated position and task point visitation status of the UGV, while v'_{task} reflects the new visitation status of all task points based on the agents' actions.

4. Reinforcement learning framework

We propose a Transformer-based RL framework for multi-UAV-UGV path planning, leveraging attention mechanisms to dynamically optimize agent trajectories and minimize mission time through an encoder-decoder architecture.

4.1. The structure of Encoder

The encoder utilizes multi-head attention to transform task point features (position, charging capability) into contextual embeddings that capture spatial relationships for subsequent decision-making.

1) **Input representation**: Each task point is represented as a vector $X_j = \{x_j, y_j, c_j\}$, where $\{x_j, y_j\}$ denotes the coordinates of the task point, and $c_j \in \{0, 1\}$ is a binary variable indicating whether the task point serves as a charging station.

2) **Initial embedding**: The input vector undergoes a linear transformation to generate the initial embedding $e_j = W_e \mathbf{x}_j + \mathbf{b}_e$, where W_e and $\mathbf{b}_e$ are trainable parameters, with the embedding dimension typically set to $d_{\text{emb}} = 128$.

3) **Multi-head self-attention layer**: The encoder comprises L stacked multi-head attention layers (e.g., $L = 3$). Each layer contains H parallel attention heads (e.g., $H = 8$). For the h-th head in a layer, the query/key/value vectors are computed as:

$$Q_h = EW_{Q_h}, \quad K_h = EW_{K_h}, \quad V_h = EW_{V_h} \tag{1}$$

where $E = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k]^\top \in \mathbb{R}^{k \times d_{\text{emb}}}$ is the embedding matrix of all k task points, and W_{Q_h}, W_{K_h}, $W_{V_h} \in \mathbb{R}^{d_{\text{emb}} \times d_h}$ are learnable weight matrices.

4) **Attention score calculation**: The scaled dot-product attention for the h-th head is computed as:

$$\alpha_{j,h} = \frac{Q_{j,h} K_{j,h}^\top}{\sqrt{d_k}} \tag{2}$$

where d_k is the dimension of key vectors, and the scaling factor $\sqrt{d_k}$ stabilizes gradient propagation.

5) **Multi-head attention output**: The outputs from all heads are concatenated and projected to form the layer output, H is the number of heads.

$$E' = \text{Concat}(V_1, V_2, \ldots, V_H) \tag{3}$$

6) **Feed-forward layer**: Each multi-head self-attention layer is followed by a feed-forward layer that uses the ReLU activation function:

$$E'' = \text{ReLU}(W_1 E' + b_1), E''' = W_2 E'' + b_2 \tag{4}$$

where W_1, b_1, W_2, b_2 are trainable parameters.

7) **Final embedding**: After passing through all the encoder layers, the final embedding $e_j^{(L)}$ for each task point is obtained. These embeddings will be passed to the decoder for subsequent decision-making.

4.2. The structure of Decoder

The decoder employs hierarchical attention to select actions by matching encoded task features with real-time states, ensuring adaptive decision-making.

1) **Construct the context vector**: For each UAV, construct the context vector $c_{UAV,i}$, which contains information about the current state; For unmanned vehicles, construct the context vector c_{UGV}, which contains the information of the current state.

2) **Multi-head attention mechanism**: For each UAV and UGV, the multi-head attention mechanism is used to calculate the similarity between the context vector and the task point embedding, thereby determining the next action. For each header h, calculate the Query, Key and Value vectors:

$$Q_{h,i} = c_{UAV,i} W_{Q_h}, \quad K_{h,j} = e_j^{(L)} W_{K_h}, \quad V_{h,j} = e_j^{(L)} W_{V_h} \tag{5}$$

where W_{Q_h}, W_{K_h}, W_{V_h} are trainable weights.

3) **Attention score calculation**: For each head h:

$$\alpha_{ijh} = \frac{Q_{h,i} K_{h,j}^{\top}}{\sqrt{d_k}} \tag{6}$$

where d_k is the dimension of key vectors for gradient stabilization.

4) **Multi-head attention output**: Concatenate the outputs of all the heads to form the multi-head attention output of this layer: $\alpha_i' = \text{Concat}(V_{i1}, V_{i2}, \ldots, V_{iH})$, where H is the number of heads

5) **Single-head attention mechanism**: The single-head attention mechanism is used to further calculate the similarity between the context vector and the task point embedding to determine the next action:

$$\alpha_{ij} = \frac{\alpha_{ij}' \cdot e_j^{(L)}}{\sqrt{d_k}} \tag{7}$$

6) **Select the action**: Based on the calculated probability distribution, select the next action. One can choose the action with the highest probability, or sample based on the probability distribution.

7) **Update status**: Update the current status to the new status based on the selected action to update the positions of the unmanned aerial vehicle and the unmanned vehicle. Update the fuel level. Update the access status of the task point

8) **Repeat the above steps**: until all task points have been visited or the task termination conditions are met.

4.3. Training method

Our framework employs the REINFORCE algorithm with baseline to train the policy network. This approach minimizes gradient variance through greedy baseline comparisons while optimizing policy parameters via reward-weighted action sampling.

Algorithm 1: Policy network training using REINFORCE algorithm

```
Input: Policy network $\pi_\theta$, Baseline network $\pi_\phi$, epochs $E$, Number of batches $N$, batch size $B$, episode length $T$
Output: Trained policy network $\pi_{\theta'}$
for epoch in 1 to $E$ do
    Sample $N$ batches from dataset
    for iteration in 1 to $N$ do
        for instance $b$ in 1 to $B$ do
            Initialize $s_{0,b}$ at $t = 0$
            while $t < T$ do
                Sample action $a_{t,b} \sim \pi_\theta(a|s_{t,b})$
                Obtain reward $r_{t,b}$ and next state $s_{t+1,b}$
                $t = t + 1$
            end
            Compute trajectory return: $\mathcal{R}_b = -\max\left\{\frac{u_a}{u_g}\right\}\sum_{t=0}^{T} r_{t,b}$
            Baseline reward $\mathcal{R}_b^{\phi}$from greedy rollout with $\pi_\phi$
        end
        Compute policy gradient: $\nabla_\theta J = \frac{1}{B}\sum_{b=1}^{B}(\mathcal{R}_b - \mathcal{R}_b^{\phi})\nabla_\theta \log \pi_\theta(s_{T,b}|s_{0,b})$
        Update parameters: $\theta \leftarrow \theta + \alpha\nabla_\theta J$
    end
    if OneSidedPairedTTest($\pi_\theta$, $\pi_\phi$) $< 0.05$ then
        $\phi \leftarrow \theta$
    end
end
```

5. RESULTS

We evaluate our framework through extensive computational experiments in an 8×8 km area with 25 task points. The setup assumes: UAVs operate at 10 m/s with 15-minute endurance;A single UGV moves at 1.5-4.5 m/s;Fixed 5-minute recharge time

The UGV path is determined via minimum set coverage and TSP solutions, while UAV routing is modeled as an energy-constrained VRPTW solved using OR-Tools. We assess performance with 3-6 UAVs under fixed mission points, analyzing the impact of fleet size on planning efficiency.

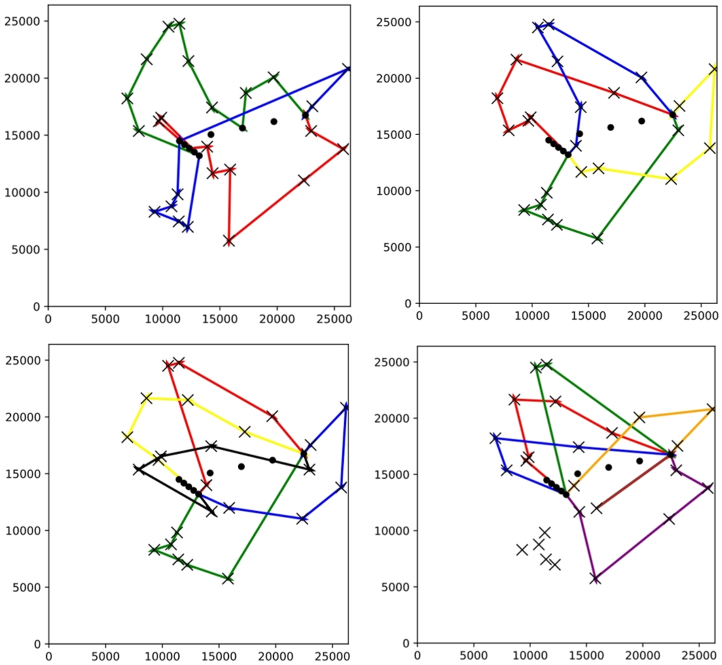


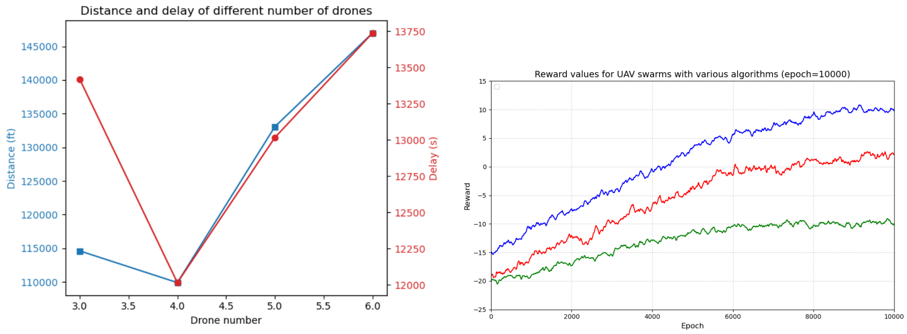
Figure 3: Results of performance comparison with different number of drones

Our experimental results (Figs. 3-4) reveal a nonlinear relationship between fleet size and system performance. The system achieves optimal efficiency at moderate UAV counts, with both mission delay and travel distance showing significant reduction. However, beyond this optimal point, further increases in fleet size lead to degraded performance in both metrics, demonstrating the inherent trade-offs in multi-agent coordination scalability.

By comparing the proposed algorithm with the baseline algorithm, its performance and effectiveness in solving the collaborative path planning of UAVs and UAVs can be evaluated. We have trained the method proposed in this paper (blue), the GLS (Guided Local Search) method (red) and the TS (Tabu Search) method (green) for 10,000 cycles in the same scenario, and the reward values under different training rounds are shown in Fig4. It can be seen that all three algorithms have converged. The reward value of the proposed algorithm is significantly higher than the other two algorithms.

6. Conclusion

In this paper, we have presented a novel Transformer-based reinforcement learning framework for UAV-UGV cooperative path planning with charging constraints. Our approach addresses three critical limitations of existing methods: the separation of path planning and charging coordination, limited scalability, and poor adaptability in dynamic environments. Through extensive experiments,



(*a*) (*b*)

Figure 4: (a) Distance and delay of different number of drones; (b) Reward value for UAV swarms with various algorithms.

we demonstrate that our framework effectively coordinates multiple energy-constrained UAVs with mobile charging UGVs, achieving superior performance compared to traditional optimization and learning-based methods. The attention mechanisms in our Transformer architecture enable efficient processing of complex agent interactions and dynamic charging requirements. Future work will focus on extending this framework to handle more diverse mission scenarios and integrating real-world communication constraints.

References

Yiqiao Cai, Zifan Lin, Meiqin Cheng, Peizhong Liu, and Ying Zhou. Solving multi-objective vehicle routing problems with time windows: A decomposition-based multiform optimization approach. *Tsinghua Science and Technology*, 29(2):305–324, 2024. doi: 10.26599/TST.2023.9010048.

Aiwen Huang, Xianger Li, Xuyang Chen, Wei Song, Zhihai Tang, Le Chang, and Tian Wang. Mobipower: Scheduling mobile charging stations for uav-mounted edge servers in internet of vehicles. *Peer-to-Peer Networking and Applications*, 18(2):82, 2025a. doi: 10.1007/s12083-025-01905-0.

Aiwen Huang, Xianger Li, Xuyang Chen, Wei Song, Zhihai Tang, Le Chang, and Tian Wang. Mobipower: Scheduling mobile charging stations for uav-mounted edge servers in internet of vehicles. *Peer-to-Peer Networking and Applications*, 18(2):82, 2025b. doi: 10.1007/s12083-025-01905-0.

Dionisis Kandris, Christos Nakas, Dimitrios Vomvas, and Grigorios Koulouras. Applications of wireless sensor networks: An up-to-date survey. *Applied System Innovation*, 3(1), 2020. doi: 10.3390/asi3010014.

Pei Li, Lingyi Wang, Wei Wu, Fuhui Zhou, Baoyun Wang, and Qihui Wu. Graph neural network-based scheduling for multi-uav-enabled communications in d2d networks. *Digital Communications and Networks*, 10(1):45–52, 2024. doi: https://doi.org/10.1016/j.dcan.2022.05.014.

Md Safwan Mondal, Subramanian Ramasamy, James D. Humann, James M. Dotterweich, Jean-Paul F. Reddinger, Marshal A. Childers, and Pranav Bhounsule. An attention-aware deep reinforcement learning framework for uav-ugv collaborative route planning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13687–13694, 2024. doi: 10.1109/IROS58592.2024.10801704.

Apiwat Nonut, Yodsadej Kanokmedhakul, Sujin Bureerat, Sumit Kumar, Ghanshyam G. Tejani, Pramin Artrit, Ali Rıza Yıldız, and Nantiwat Pholdee and. A small fixed-wing uav system identification using metaheuristics. *Cogent Engineering*, 9(1):2114196, 2022. doi: 10.1080/23311916.2022.2114196.

Saroj Paudel, Jiangfeng Zhang, Beshah Ayalew, and Annette Skowronska. Charging load estimation for a fleet of autonomous vehicles. In *WCX SAE World Congress Experience*. SAE International, April 2024a. doi: https://doi.org/10.4271/2024-01-2025.

Saroj Paudel, Jiangfeng Zhang, Beshah Ayalew, and Annette Skowronska. Charging load estimation for a fleet of autonomous vehicles. pages 2024–01–2025. SAE International, apr 2024b.

Christian Tilk, Nicola Bianchessi, Michael Drexl, Stefan Irnich, and Frank Meisel. Branch-and-price-and-cut for the active-passive vehicle-routing problem. *Transportation Science*, 52(2):300–319, 2018. doi: 10.1287/trsc.2016.0730.

Xiuzhu Zhang. Path planning and control of intelligent delivery uav based on internet of things and edge computing. *International Journal of Advanced Computer Science and Applications*, 15(3), 2024. doi: 10.14569/IJACSA.2024.01503107.

Zhiming Zhou, Hongmin Qi, Zhen Liu, and Dianwei Qian. Research on autonomous decision-making of multi-uav air combat based on deep reinforcement learning. Proceedings of 2023 7th Chinese Conference on Swarm Intelligence and Cooperative Control, pages 687–700. Springer Nature Singapore.