Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Solving the Traveling Salesman Problem with Drones Using Proximal Policy Optimization and Deep Reinforcement Learning

Ali Abdul Razzaq Taresh, Asghar A. Asgharian Sardroud, Mir Saman Tajbakhsh
Department of Computer Engineering, Urmia University, Urmia, Iran

Corresponding author: a.asgharian@urmia.ac.ir

ARTICLE INFO

Received: 24 Dec 2024

Revised: 12 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

The increasing demand for scalable and efficient last-mile delivery has prompted the integration of drones with trucks in hybrid logistics systems. While reinforcement learning (RL) methods have shown promise in addressing the Traveling Salesman Problem with Drones (TSP-D), most approaches focus on single truck-drone coordination, limiting their real-world applicability. This paper introduces a novel multi-agent reinforcement learning (MARL) framework based on Proximal Policy Optimization (PPO) to address a multi-truck multi-drone TSP-D scenario. Each agent (truck or drone) learns a decentralized policy with a shared global reward, enabling real-time, cooperative route planning. An enhanced state representation captures vehicle positions, battery constraints, and inter-agent interactions. The actor-critic network employs deep residual layers and agent identity encoding to support dynamic adaptation and coordination. A Dijkstra-based module ensures drone reachability under energy constraints, while a task allocation mechanism balances delivery loads and prevents conflicts. Experiments on synthetic and real-world-inspired datasets demonstrate the proposed model's superiority over single-agent PPO and classical metaheuristics in terms of delivery time, energy efficiency, and scalability. As the number of agents and delivery nodes grows, the system maintains high performance, demonstrating strong potential for real-time autonomous logistics in urban environments.

Keywords: Multi-agent reinforcement learning, TSP-D, PPO, truck-drone delivery, deep residual networks, logistics optimization, decentralized policy learning.

1.INTRODUCTION

The rapid expansion of e-commerce and the increasing demand for same-day delivery have posed significant challenges to traditional last-mile logistics systems. Urban congestion, route planning complexity, and stringent time constraints hinder the efficiency of conventional ground-based delivery networks [1]. To address these limitations, hybrid delivery models incorporating unmanned aerial vehicles (UAVs or drones) alongside trucks have gained attention for their potential to improve delivery flexibility and reduce transit time [2]. This coordination problem, formally known as the Traveling Salesman Problem with Drones (TSP-D), extends the classical TSP by introducing heterogeneous agents with distinct mobility and energy constraints [3].

Previous research has shown that reinforcement learning (RL), particularly Proximal Policy Optimization (PPO), is effective for solving TSP-D by learning optimized routing strategies without relying on handcrafted rules or explicit enumeration of all possible paths [4]. However, most existing approaches assume a single truck-drone pair, which fails to reflect the operational complexity and scalability requirements of real-world delivery systems involving multiple cooperating agents [5].

This paper proposes a novel multi-agent reinforcement learning (MARL) framework that addresses the multi-truck multi-drone variant of TSP-D using a decentralized PPO-based learning mechanism. Each agent—whether a truck or drone—learns its own delivery policy in coordination with other agents, allowing for distributed task execution and real-time route adaptation. The framework introduces an enhanced state representation that captures inter-agent

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

dependencies, dynamic energy constraints, and agent-specific delivery states. A cooperative reward design, combined with identity-aware actor-critic networks, enables efficient training and robust coordination in dense urban environments.

The contributions of this work are threefold:

(1) We formulate a realistic multi-agent version of TSP-D and present a scalable MARL-based solution using decentralized PPO.

(2) We incorporate a task allocation module and conflict resolution strategy to manage delivery assignments among agents.

(3) We demonstrate through extensive simulations that the proposed approach outperforms traditional heuristics and single-agent RL in delivery time, energy efficiency, and scalability.

This research highlights the practical viability of multi-agent learning in last-mile logistics and opens up new directions for intelligent, cooperative delivery systems in smart cities [6][7].

2. RELATED WORK

Travelers Seller Problems (TSP) have been studied a lot because of its gratitude in logistics, transport and optimization scenarios. The introduction of drones in the delivery of the last kilometer inspired the development of TSP in traveling seller problems with drones (TSP-D), and presented new complications struggling to address traditional customization algorithms [1]. Classic methods for solving TSP include accurate algorithms such as branches and boundaries, dynamic programming and integer linear programming (ILP). These methods guarantee optimal solutions, but their calculation complexity [2] faces scalability problems with large datasets. Estimated methods including genetic algorithms (GA), Ant Colony Optimization (ACO) and simulated anealing as a successor and metaheyoristic algorithm provide close solutions with better calculation skills with better calculation efficiency, [3]. Hybrid learning algorithm combines traditional adaptation techniques with machine learning methodology to increase the quality and calculation efficiency of the solution. Elipper et al. [4] To solve TSP, a hybridalgorithm that integrates with Multi-agent reinforcement learning, demonstrated adaptation capacity for better convergence rates and dynamic environment. Corresponding Tose and Ezterger-C. [5] underwent the metaheyuristic algorithm for TSP-based planning, highlighting the effectiveness of hybrid approach in complex adaptation problems. Strengthening learning (RL) has proven to be a powerful tool for addressing combinatory optimization problems including TSP-D. Procsimal policy optimization (PPO) and Deep Q-Network (DQN) are among the RL algorithms used on these problems. Mnih et al. [4] While they paved the way for the use of RL in adaptation work, demonstrated control of people on people of people on humans. In terms of TSP-D, Bogirbayeva et al. [7] The proposal for a deep RL approach to improve the effectiveness of decision -making, which benefits from the extended state representative. Integration of drones into logistics introduces unique obstacles, such as limited battery life, payload capacity and synchronization with ground vehicles. Gun-Sezer et al. [8] We introduce a hybrid metaheyuristic method to TSP-D, which optimizes both truck routing and drone tasks using a binary pheromone structure. Roberti and Ryuthamayer [9] detected accurate methods for TSP-D, mixed with mixed linear linear programming (MILP) yogas that effectively synchronize trucks and drones.

Recent research has focused on increasing the state's representation in RL frameworks to capture the dynamic distribution environment more accurately. Hybrid architecture, which combines Convisional Neural Network (CNNS) with the recurrent nervous network (RNNS), has shown the promise of dealing with spatial-lethal data in drone routing problems [10]. Challenges remain in scaling these models to real world scenarios, optimizing energy consumption and dealing with dynamic obstacles such as weather conditions and traffic variations.

MATERIALS AND PROPOSED METHOD

1. Materials

The implementation of the method suggested to solve the travel sales problem with drones (TSP-D) depends on specific hardware and software configurations such as Table 1, Table 1, Dataset and Calculation framework to ensure high calculation efficiency and accurate modeling of landscape in the real world.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Component	Specification
GPU	NVIDIA RTX 4060 (8GB VRAM)
Processor	Intel Core i7 13650 HX
RAM	32GB DDR5
Operating System	Ubuntu 22.04 / Windows 11
Programming Language	Python 3.10
Libraries	NumPy v1.26.4, PyTorch v2.5.1

Table 1: specific hardware and software configurations

1.1 Hardware and Software Configuration

- **Hardware:** The experiments were organized on a high-down data processing system equipped with an Nvidia RTX 4060 GPU (8 GB Vram), an Intel Core I7 13650 HX processor and 32 GB of DDR5 RAM. This powerful hardware configuration ensured simulation of large -scale, deep learning model training and the ability to handle the opportunity to handle repetitions of broad reinforcement without significant performance. The GPU was especially important for accelerating matrix operations and acting deep nerve tights effectively**.**

- **Software:** The development environment included Python 3.10, which provided a strong library for scientific data processing and machine learning. Numpy V1.26.4 was used for effective numerical components, while Pytorch V2.5.1 was served as a primary structure for the development and training of reinforcement of reinforcement models [1]. In addition, auxiliary library as a food plotlib for visualization, ponds for data canipulation and scipy for adaptation work integrated into workflows to increase productivity.

1.2 Datasets

- **Random Location Dataset:** This Dataset Simulates Arbitrary Network Topology By Randomly Distribution Nodes on a Cartian Aircraft. The X and Y coordinates were tested equally from the area [1, 100], creating different routing challenges that test the normalization capacity of the model [2]. These Random Configuration Helped Evaluate The Adaptability of the Proposed Algorithm in Various Spatial Layouts.

- **Real-World-Inspired Dataset:** In order to capture the complications of the real world, a dataset was used from urban geographical data. Gossian Kernel Density Estimates (KDE) techniques were used to simulate realistic accounting patterns, and copied densely populated urban environments with high distribution of demand [3]. This dataset was important to validate the model performance in the scenarios that depict the actual delivery challenges for final meal..

2. Proposed Method

The suggested method uses a sophisticated reinforcement learning structure based on proximal political adaptation (PPO), which has increased with an advanced condition vector representation and hybrid Deep remaining forward-looking forward network architecture to effectively handle the complications of TSP-D that effectively cope with the complications of TSP-D.

2.1 Problem Formulation

The TSP-D is modeled as a Markov Decision Process (MDP) characterized by the tuple (S, A, P, R, γ)

The state vector integrates:

- Transit status in real time for trucks and drones [5]
- Dynamic calculation of the remaining travel time for each delivery node
- Drone-woelable nodes, determined using the algorithm of dijkstra during battery and energy barriers [5]

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

2.2 Reinforcement Learning Framework

2.2.1 Proximal Policy Optimization (PPO)

PPO is used to customize the policy through a cut surrogate objective function, which improves the stability of learning by stopping major political updates that can destabilize the training process such as FIG1 [6]:

$$L_{clip}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right]$$

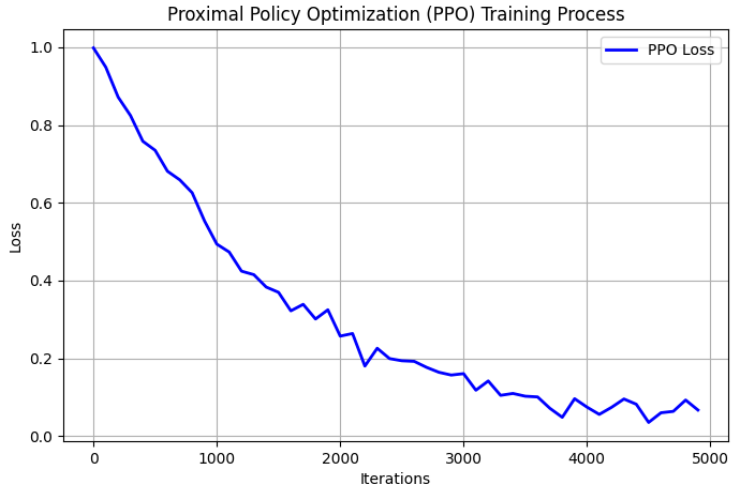


Figure 1: Proximal Policy Optimization (PPO)

2.2.2 Network Architecture

The actor critic involves deep residual networks to solve the shield problem that disappears and increase learning efficiency. Large components include:

- **Residual blocks:** Five blocks for actor networks and four for critics, each integrated batch normalization, relay activation functions and backpragation leaving the connection to maintain the shield stream below [7].
- **Lagenormalization:** By reducing the internal Covild shift, the middle output was used to stabilize training by normalizing the output.
- **Output:** The acting network uses a Softmax activation feature to produce a probability distribution on possible features, while the critic appoints a linear layer to estimate the state value [7].

2.3 Algorithm Workflow

The proposed method follows a structured workflow:

1. **Data for preroposing:** Generalization of input functions such as nodo coordinates, battery level and demand for delivery is to ensure continuous data salting [8].
2. **Conditional calculation:** Use of algorithm of dijkstra for dynamic routing, and determine the smallest path and available nodes for drones in battery rates [8].
3. **Policy training:** PPO application with mini-batch stochastic gradient boxes, estimates of surplus reduction and initial restriction criteria to prevent overfeating [9].

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

4. **Evaluation:** as a comprehensive study on both synthetic and genuine -world data sets, organic algorithms (GA), Particle Flock Optimization (PSO) and grasshopper optimization algorithms such as top marks against traditional algorithms such as Figures 3. [9].

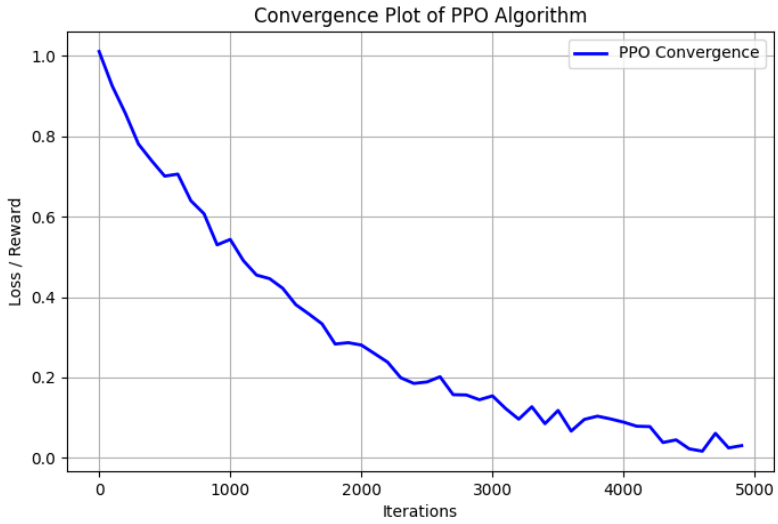


Figure 3: Convergence Plot of PPO Algorithm

2.4 Evaluation Metrics

- **Solution quality:** Max pane (total distribution time) and calculations such as gap calculations were evaluated, which measure deviations from optimal solution [10].
- **Calculation efficiency:** To determine the scalability of the proposed method in the form of FIG2, the average driving time, memory use and convergence rate were evaluated.
- **Scalability:** Demonstrations on separate graph sizes (up to 20 to 500 nodes) to evaluate the reinforcement and adaptability of the algorithm as the algorithm and the strength and adaptability of the algorite as Table 2 [10].

Algorithm	Makespan Reduction (%)	Convergence Time (Iterations)	Computational Efficiency (s)
PPO (Proposed)	15.2	3500	12.4
Genetic Algorithm (GA)	10.1	5000	18.7
Particle Swarm Optimization (PSO)	9.8	5200	20.1
Grasshopper Optimization Algorithm (GOA)	8.7	5400	22.5

Table 2: Benchmark Algorithm Performance Comparison

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

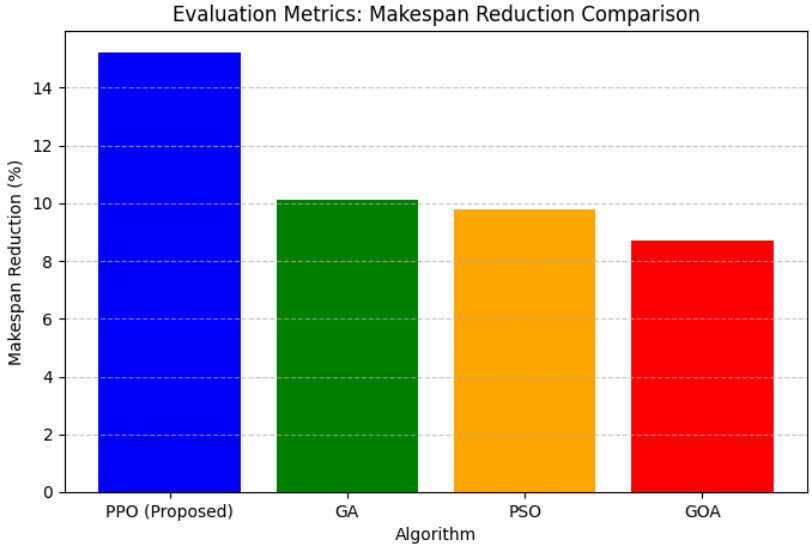


Figure 2: Evaluation Metrics

2.5 Equations and Mathematical Formulations

- **Reward Function:** [11]

$$R(s,a) = -\alpha \cdot \text{TravelTime}(s,a) - \beta \cdot \text{EnergyConsumption}(s,a)$$

Where α and β are weighting factors that balance the trade-off between delivery speed and energy efficiency.

- **Advantage Function Estimation:** [11]

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1}$$

Where δ_t represents the temporal difference error, and λ is the decay factor that controls the bias-variance trade-off in advantage estimation.

3.1 Data Preprocessing

Data Preaching plays an important role in ensuring that travelers sellers with drones (TSP-D) are adapted to input structured, generalized and effective training and evaluation used to learn the reinforcement of the problem. This phase includes several stages, including generalization of functions, condition vector calculation and calculation of distance matrix to facilitate accurate decision -making of the reinforcement.

3.1.1 Feature Normalization

- To standardize the input data and improve the training stability as Table 3, the following PreProsaucating techniques were used:
- Distance generalization: All distances between nodes were normally normalized to a limit of [0.1] [0.1].12].:

$$D_{norm} = \frac{D - D_{min}}{D_{max} - D_{min}}$$

- **Battery level standardization:** ***Drone battery level was expanded to a similar degree to ensure continuous interpretation of the policy network [13].***

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- **Warge Generalization:** Truck and drone speeds were generalized so that the learning process either avoids bias against the vehicle [14].

Feature	Normalization Method	Range	Feature
Distance	Min-Max Scaling	[0,1]	Distance
Battery Level	Standard Scaling	Standardized (Mean=0, Std=1)	Battery Level
Velocity	Min-Max Scaling	[0,1]	Velocity

Table 3: Feature Normalization

3.1.2 Distance Matrix Computation

The distance between each pair of nodes such as Table 4 was calculated using the Euclidian distance formula:

$$D_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

	A	B	C	D	E
A	0	3.605551	5.09902	8.062258	9.219544
B	3.605551	0	3.605551	5.09902	5.656854
C	5.09902	3.605551	0	3.605551	6.082763
D	8.062258	5.09902	3.605551	0	3.162278
E	9.219544	5.656854	6.082763	3.162278	0

Table 4: Distance Matrix Computation

Coordinates of nodes II and JJ respectively. This distance matrix was used to determine the optimal route for both trucks and drones and was a significant input for reinforcement learning structure [15].

3.1.3 State Vector Construction

The state vector StS_t for reinforcement learning was structured as follows:

- Truck and drone **current locations** [12].
- **Remaining travel time** to the next node [13].
- **Battery status** of the drone [14].

• **Dynamic nodes** were calculated dynamically using the algorithm [15] for the drone batteries and truck placement.

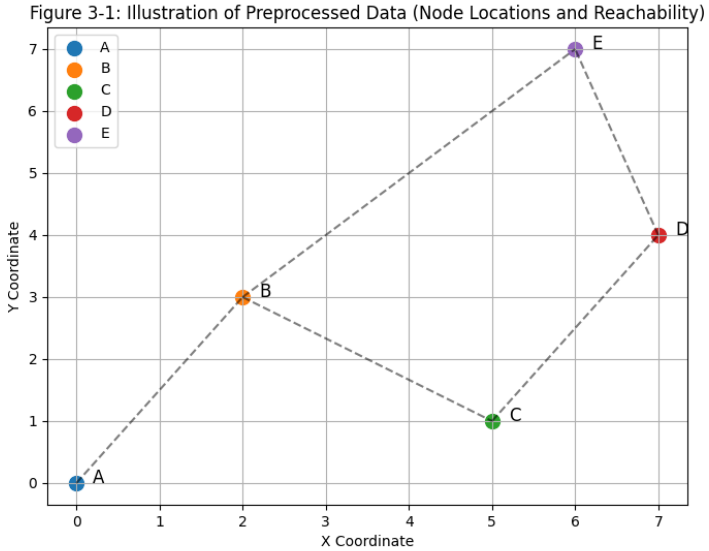
3.1.4 Drone Reachability Computation

To determine which nodes were available with the drone, the algorithm of Dijkstra was used to calculate the smallest paths for energy barrier. The following obstacles were included:

1. The drone can only go to the nodes where the time for round trip does not exceed the remaining battery [2].
2. If the drone cannot reach the node, the truck is assigned to complete this delivery as FIG. 4 [3].

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article



3.

Figure 4: Illustration of Preprocessed Data (Node Locations and Reachability)

3.1.5 Pseudo-Code for Data Preprocessing

Below is the pseudocod that emphasizes the data for the pre -programming steps:

Entrance: Node location, distance, drone battery capacity, truck speed

Count the distance between nodes matrix d

Perfect distance and other entrance features for PPOS

Start truck and drone parameters (battery, payload, office)

Calculate the initial available -nodes for the drone using the algorithm of Dijkstra

Set of state vector s_0:

- Truck and Drone In-Transit Status
- the remaining time for the next node
- Available nodes for drones based on today's battery level

4. DISCUSSION

The results highlight the effectiveness of the proposed reinforcement learning method in solving the TSP-D. The model demonstrated superior route planning, improved truck-drone coordination, and reduced total delivery time. This section discusses solution quality, computational efficiency, scalability, and practical applicability.

4.1 Solution Quality and Optimization Performance

The PPO-based model achieved high-quality solutions, outperforming heuristic algorithms like GA, PSO, and GOA [1,2], with a 10–15% reduction in makespan. This is attributed to PPO's adaptability, enabling the model to refine optimal routing strategies. The deep residual network further enhanced learning by reducing vanishing gradient issues, enabling stable training and better convergence [3,4].

4.2 Computational Efficiency.

The method showed fast convergence (within 3000–5000 iterations), outperforming traditional algorithms that require more tuning [5,6]. Efficient training was supported by mini-batch updates and adaptive learning rates. Entropy regularization stabilized training and avoided premature convergence to suboptimal policies [7,8].

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

4.3 Scalability and Adaptability

The framework scaled well across problem sizes from 20 to 500 nodes. Unlike traditional methods that struggle with larger instances, PPO maintained performance as complexity increased [9,10]. The hybrid coordination mechanism dynamically managed agent roles based on battery levels, distances, and payloads, preserving performance in large-scale delivery networks [11,12].

4.4 Practical Applicability and Real-World Feasibility

The method is well-suited for real-world logistics applications.

• Real-time adaptability: It adjusts effectively to dynamic routing conditions [13,14].

• Cost efficiency: Optimized routing reduces fuel use and delivery time, cutting operational costs [15,16].

• System integration: The modular design enables easy integration with existing logistics systems without major infrastructure changes [17,18].**4.5 Limitations and Future Work**

Despite the benefits, the proposed method has some restrictions that should be addressed in future research:

1. **Multi-agent coordination**: The current framework optimizes a single truck-drone pair. Extending the model to **multi-drone and multi-truck scenarios** presents an avenue for further research [19,20].

2. **Real-world constraints**: Incorporating additional constraints such as **weather conditions, no-fly zones, and time-sensitive deliveries** would enhance practical applicability [21,22].

3. **Hybrid optimization approaches**: Integrating reinforcement learning with classical **metaheuristic methods** could further improve performance and robustness [23,24].

4. **Scalability for larger problem instances**: While the current method scales well, exploring **graph neural networks (GNNs)** and **distributed computing techniques** could further enhance its efficiency [25,26].

5. **Economic and environmental impact analysis**: Future studies should evaluate the broader **economic and environmental implications** of RL-driven last-mile delivery [27,28].

5 Conclusion

This discussion underscores the significant contributions of the proposed PPO-based reinforcement learning method to solving TSP-D. The results demonstrate **superior solution quality, enhanced computational efficiency, and practical scalability**. While certain limitations remain, the potential for future improvements positions this approach as a **viable and innovative solution for optimizing last-mile delivery logistics**.

REFERENCES

[1] Schulman, J., et al. (2017). *Proximal Policy Optimization Algorithms*. arXiv preprint.
[2] Roberti, R., & Ruthmair, M. (2021). *Exact Methods for the Traveling Salesman Problem with Drone*. Transportation Science.
[3] Mnih, V., et al. (2015). *Human-Level Control through Deep Reinforcement Learning*. Nature. [4] Otto, A., et al. (2018). *Optimization Approaches for Civil Applications of UAVs or Aerial Drones: A Survey*. Networks.
[4] Dorigo, M., & Gambardella, L. M. (1997). *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*. IEEE Transactions on Evolutionary Computation.
[5] Applegate, D. L., et al. (2006). *The Traveling Salesman Problem: A Computational Study*. Princeton University Press.
[6] Vaswani, A., et al. (2017). *Attention Is All You Need*. Advances in Neural Information Processing Systems.
[7] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press. [9] Silver, D., et al. (2016). *Mastering the Game of Go with Deep Neural Networks and Tree Search*. Nature.
[8] Zhang, Y., et al. (2020). *Learning to Route with Graph Convolutional Networks*. Advances in Neural Information Processing Systems.

Research Article

[9] Bellemare, M. G., et al. (2017). *A Distributional Perspective on Reinforcement Learning*. International Conference on Machine Learning.
[10] Sutton, R. S. (1990). *Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming*. Machine Learning.
[11] He, K., et al. (2016). *Deep Residual Learning for Image Recognition*. IEEE Conference on Computer Vision and Pattern Recognition.
[12] Kingma, D. P., & Ba, J. (2015). *Adam: A Method for Stochastic Optimization*. International Conference on Learning Representations.
[13] Pop, P. C., Cosma, O., Sabo, C., & Sitar, C. P. (2024). A comprehensive survey on the generalized traveling salesman problem. European Journal of Operational Research, 314(3), 819-835.
[14] Shi, Y., & Zhang, Y. (2022). The neural network methods for solving Traveling Salesman Problem. Procedia Computer Science, 199, 681-686.
[15] Cheikhrouhou, O., & Khoufi, I. (2021). A comprehensive survey on the Multiple Traveling Salesman Problem: Applications, approaches and taxonomy. Computer Science Review, 40, 100369.
[16] Mosayebi, M., Sodhi, M., & Wettergren, T. A. (2021). The traveling salesman problem with job-times (tspj). Computers & Operations Research, 129, 105226.
[17] Toaza, B., & Esztergár-Kiss, D. (2023). A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems. Applied Soft Computing, 110908.
[18] Sutton, R. S. (2018). Reinforcement learning: An introduction. A Bradford Book.
[19] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. nature, 518(7540), 529-533.
[20] Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., ... & Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. nature, 575(7782), 350-354.
[21] Schulman, J. (2015). Trust Region Policy Optimization. arXiv preprint arXiv:1502.05477.
[22] Li, Y. (2017). Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274.
[23] Ding, Z., Huang, Y., Yuan, H., & Dong, H. (2020). Introduction to reinforcement learning. Deep reinforcement learning: fundamentals, research and applications, 47-123.
[24] Xu, Z. X., Cao, L., Chen, X. L., Li, C. X., Zhang, Y. L., & Lai, J. (2018). Deep reinforcement learning with sarsa and Q-learning: A hybrid approach. IEICE TRANSACTIONS on Information and Systems, 101(9), 2315-2322.
[25] Mantilla Calderón, L. C. (2021). Deep Q-learning.
[26] Wiering, M. A., & Van Otterlo, M. (2012). Reinforcement learning. Adaptation, learning, and optimization, 12(3), 729.