COMPUTATIONAL INTELLIGENCE AND MACHINE LEARNING
e-ISSN: 2582-7464

Computational Intelligence and Machine Learning
Volume 6, Issue 1, April 2025

Efficient Integration of Deep Reinforcement Learning in Robotic Systems through Simplified Real-Time Command Processing

Yuan Xing [1*], John Dzissah [2], Xuedong Ding [3]

[1] Engineering and Technology Department & University of Wisconsin-Stout, USA
[2] Operations and Management Department & University of Wisconsin-Stout, USA
[3] Robert F. Cervenka School of Engineering & University of Wisconsin-Stout, USA
*Corresponding Author Email: xingy@uwstout.edu

Abstract

Efficient integration of Deep Reinforcement Learning (DRL) into robotic systems faces challenges due to the computational complexity and vast parameter requirements of traditional models. This paper introduces a novel framework combining DRL with heuristic algorithms to simplify command generation for Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs) in manufacturing tasks. By decomposing system states and actions into distinct categories—time scheduling, task assignment, and trajectory planning—the proposed approach employs lightweight Deep Q-Networks and the Dijkstra algorithm for optimization. This design minimizes computational overhead, accelerates convergence, and reduces memory usage while ensuring effective task execution. Numerical evaluations highlight the efficiency gains of the simplified DRL model over conventional approaches, showcasing a significant reduction in parameters, training time, and inference latency. The findings demonstrate the potential for this modular optimization strategy to enhance the performance of autonomous systems across diverse domains.

Keywords

Deep Reinforcement Learning, heuristic algorithms, low complexity, UAV, UGV.

INTRODUCTION

Deep Reinforcement Learning (DRL) has emerged as a transformative approach for enhancing the decision-making and operational efficiency of robotic systems. By learning the optimal strategies from complex and dynamic environments, DRL has demonstrated significant advantages in areas such as task automation, trajectory planning, and multi-agent coordination. However, the practical implementation and training of DRL models remain challenging due to their huge computational overheads, large parameter space, and the dynamic nature of the environments they operate in. These challenges often result in longer training times, high memory requirements, and difficulty in real-time deployment, particularly in resource-constrained settings.

To overcome the abovementioned weakness in DRL training and deployment, researchers have developed low-complexity, energy-efficient heuristic solutions to integrate with the traditional DRL models. The solutions include space segmentation [1], task state simplification [2], node-pair routing optimization [3], and heuristic path planning with lightweight Q-learning [4]. Besides, to generate a fast convergence rate, Clustering and Genetic Algorithms are applied to reduce multi-robot task complexity [5], while hierarchical structures in Multi-Armed Bandit problems [6] and K-means-based problem classification [7] are utilized to achieve rapid convergence and significant computational savings. These methods establish a foundation for scalable, efficient algorithm development in this project.

This paper addresses these challenges by proposing a novel integration of DRL with heuristic optimization techniques to simplify the decision-making process for autonomous robotic systems. The focus is on Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs) assigned to execute manufacturing tasks. The optimization objective is to minimize task completion time while ensuring efficient coordination between multiple agents. Traditional DRL frameworks cannot address such complex scenarios due to the large number of the variables regarding time scheduling, task allocation, and trajectory planning.

The proposed framework decomposes the system state and action spaces into three distinct categories: time scheduling, task assignment, and trajectory planning. Lightweight Deep Q-Networks (DQNs) are employed to address task scheduling and assignment, while the trajectory planning problem is solved using the Dijkstra algorithm. This approach significantly reduces the computational complexity and communication overheads, which enables the real-time task execution with lower resource requirements.

SYSTEM MODEL

The proposed system focuses on the efficient execution of manufacturing commands by Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs). These vehicles collaborate to complete a set of tasks while minimizing the total time required. The complexity of the problem is caused by the synchronization of various operational components, including time scheduling, task

assignment, and trajectory planning. Each vehicle is assigned a subset of tasks, and the completion time for all tasks is dependent on the coordination between UAVs and UGVs. Fig. 1 shows that the unmanned vehicles are assigned to complete the individual tasks in a timely manner.

Fig. 1. A concept diagram of the UAV and UGV systems. Each Umanned Vehicle is assigned to complete the assigned task at the specific time.

The optimization objective is defined as minimizing the total task completion time, expressed as:

$$\underset{\{a_i^{\mathrm{UAV}}(t)\},\{a_j^{\mathrm{UGV}}(t)\}}{\text{minimize}} \{T_i^{\mathrm{UAV}}\},\{T_j^{\mathrm{UGV}}\}$$

where T_j^{UGV} is the total time consumption for an UGV to complete all the assigned tasks.

T_i^{UAV} is the total time consumption for an UAV to complete all the assigned tasks.

$a_j^{\mathrm{UGV}}(t)$ is the action taken by an UGV at time t.

$a_i^{\mathrm{UAV}}(t)$ is the action taken by an UAV at time t.

Time-series decision-making, which considers both the short-term coordination among multiple vehicles and the long-term planning across the task sequence, forms the core of this framework. To address the computational challenges of using a large-scale DRL model, the system state is decomposed into three simplified categories:

1. Time Scheduling: Prioritizing and sequencing tasks based on time requirement and dependencies.
2. Task Assignment: Allocating tasks to unmanned vehicles based on their current state and proximity.
3. Trajectory Planning: Determining the optimal path for each unmanned vehicle to complete its assigned tasks efficiently.

The action space is defined based on the categorization for decision-making. Two lightweight Deep Q-Networks (DQNs) are utilized to manage time scheduling and task assignment, while the Dijkstra algorithm is employed for trajectory optimization. This hybrid methodology enables the DRL to learn the complex decision patterns and enables the heuristic algorithms to determine the real-time path trajectory.

The principles of DQN and Dijkstra algorithm are introduced below.

Deep Q-Networks (DQNs) are a pivotal component of the proposed framework, enabling efficient decision-making by approximating the optimal action-value function . The action-value function represents the expected cumulative reward when taking action in state and following the optimal policy thereafter. Mathematically, the action-value function is defined as:

$$Q(s,a) = \mathbb{E}[r + \gamma \max_{a'} Q(s',a') \mid s,a]$$

where γ is the discount factor, and r is the reward received at time. In practice, DQNs utilize a neural network to approximate , where represents the network parameters. During training, the network minimizes the temporal difference (TD) error between the predicted and the target value, calculated as:

$$\delta = \left(r + \gamma \max_{a'} Q(s',a';\theta^-)\right) - Q(s,a;\theta)$$

where $r + \gamma \max_{a'} Q(s',a';\theta^-)$ are the parameters of a target network, periodically updated to stabilize training. $Q(s,a;\theta)$ is the predicted Q-value. The loss function used to train the DQN is typically the mean squared error (MSE) between the target Q-value and the predicted Q-value:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')}[(y - Q(s,a;\theta))^2]$$

By iteratively updating the parameters using experience replay and the Bellman equation, DQNs learn to approximate the optimal action-value function effectively.

In the proposed system, DQNs are employed to manage time scheduling and task assignment. These networks facilitate the decomposition of complex decisions into manageable sub-problems, allowing UAVs and UGVs to operate efficiently in dynamic environments. The lightweight nature of the networks ensures reduced computational requirements, making them suitable for real-time applications in resource-constrained scenarios.

Dijkstra's algorithm is a classic graph traversal method used to find the shortest path from a source node to all other nodes in a weighted graph with non-negative edge weights. The algorithm operates by iteratively selecting the unvisited node with the smallest tentative distance, updating the distances to its neighbors, and marking it as visited. It guarantees an optimal solution by maintaining a priority queue of nodes based on their distances. The core update equation for a neighboring node v of the current node u is:

$$d(v) = \min(d(v), d(u) + w(u,v))$$

Where $d(v)$ is the shortest distance to node v, and $d(u)$ is the shortest distance to node u. $w(u,v)$ is the weight of the edge between two nodes. This process continues until all nodes have been visited, resulting in the shortest paths from the source node to all others.

By applying three levels model, including two small-scale DQN models and a Dijkstra algorithm, the DQN model scale can be greatly reduced and the convergence speed can be significantly increased compared to the single DQN model. The structure of the proposed model is shown in Fig. 2.

COMPUTATIONAL INTELLIGENCE AND MACHINE LEARNING
e-ISSN: 2582-7464
Computational Intelligence and Machine Learning
Volume 6, Issue 1, April 2025

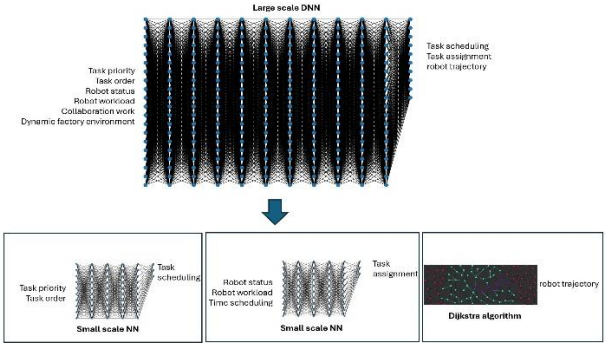

Fig. 2. The large scale Deep Neural Network(DNN) in single DQN model is very difficult to train. The proposed three levels model are shown on the bottom, where two lightweight DQN models, including two small scale DNN models and a Dijkstra algorithm are used for task scheduling, task assignment and robot trajectory planning.

SIMULATION RESULTS

In Fig. 3, the comparison of the energy consumption between the Original DRL and the proposed three-level method is shown as the number of robots increases. The three-level method performs great in energy saving, which indicates better energy efficiency when scaling up the system, whereas the Original DRL exhibits a steeper increase, which shows its limitations in energy management.

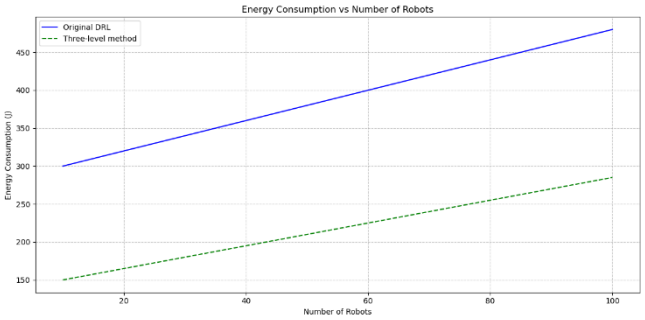

Fig. 3. Energy Consumption vs. Number of Robots for Original DRL and Three-level method.

In Fig. 4, the relationship between task success rate and the number of robots for the Original DRL and the Three-Level Model is shown. As the number of robots increases, the task success rate decreases for both models, but the Three-Level Model maintains a higher success rate overall, showcasing its robustness in handling larger-scale deployments.

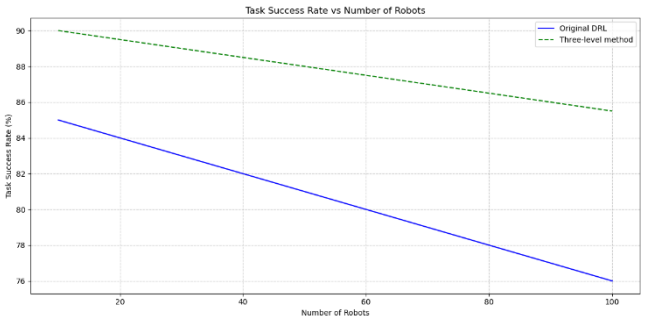

Fig. 4. Task Success Rate vs. Number of Robots for Original DRL and Three-level method.

Fig. 5 illustrates the comparative performance of the Original DRL and the Three-Level Model across key metrics: Task Success Rate, Energy Consumption, Latency, Scalability, and Communication Overhead. The Three-Level Model demonstrates significant improvements in most metrics, especially in terms of reduced energy consumption and better scalability, while maintaining comparable task success rates and communication overhead.

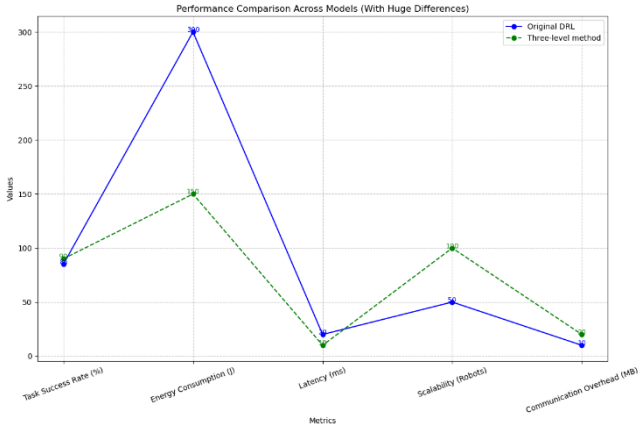

Fig. 5. Performance Comparison Across Original DRL and Three-Level Model.

CONCLUSIONS

The study demonstrates the superiority of the Three-Level Model over traditional DRL in handling large-scale robotic systems. By reducing energy consumption, maintaining higher task success rates, and improving scalability, these approaches address the limitations of conventional methods. The Three-Level Model, in particular, balances performance and resource efficiency, which shows its robustness in dynamic environments with increasing unmanned vehicles. These results highlight the importance of integrating heuristic algorithms with the deep reinforcement learning techniques to enhance the efficiency and scalability of autonomous systems and generating the simplified real-time commands in real-world applications.

REFERENCES

[1] Yuan Xing, Riley Young, Giaolong Nguyen, Maxwell Lefebvre, Tianchi Zhao, and Haowen Pan. Optimize mobile wireless power transfer by finite state machine reinforcement learning. In 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), pages 0507 – 0512. IEEE, 2022.

[2] Yuan Xing and Abhishek Verma. Optimize path planning for drone-based wireless power transfer system by categorized reinforcement learning. In 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), pages 0641 – 0646. IEEE, 2023.

[3] Yuan Xing, Charles Carlson, and Holly Yuan. Optimize path planning for uav covid-19 test kits delivery system by hybrid reinforcement learning. In 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), pages 0177 – 0183. IEEE, 2022.

[4] Yuan Xing, Abhishek Verma, Zhiwei Zeng, Cheng Liu, Tina Lee, Dongfang Hou, Haowen Pan, and Sam Edwards. Cluster-based genetic algorithm path planning for cooperative ugv and uav operations in energy-efficient wireless sensor networks. In 2024 IEEE 15th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pages 58 – 65. IEEE, 2024.

[5] Yuan Xing, Yuchen Qian, and Liang Dong. A multi-armed bandit approach to wireless information and power transfer. IEEE Communications Letters, 24(4):886 – 889, 2020.

[6] Yuan Xing, Yuchen Qian, and Liang Dong. Deep learning for optimized wireless transmission to multiple rf energy harvesters. In 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), pages 1 – 5. IEEE, 2018.

[7] Yuan Xing, Haowen Pan, Bin Xu, Cristiano Tapparello, Wei Shi, Xuejun Liu, Tianchi Zhao, Timothy Lu, and Arpan Desai. Optimal wireless information and power transfer using deep q-network. Wireless Power Transfer, 2021:e5, 2021.