Autonomous Aerial Mobility Learning for Drone-Taxi Flight Control

°Won Joon Yun, °Yoo Jeong Ha, †Soyi Jung, and °Joongheon Kim
°School of Electrical Engineering, Korea University, Seoul, Republic of Korea
†School of Software, Hallym University, Chuncheon, Republic of Korea
joongheon@korea.ac.kr

2021 International Conference on Information and Communication Technology Convergence (ICTC) | 978-1-6654-2383-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICTC52510.2021.9620751

***Abstract*—In smart city scenarios, the use of unmanned aerial vehicle (UAV) networks is one of actively discussed technologies. In this paper, we consider the scenario where carpoolable UAV-based drone taxis configure their optimal routes to deliver packages and passengers in an autonomous and efficient way. In order to realize this application with drone-taxi UAV networks, a multi-agent deep reinforcement learning (MADRL) based algorithm is designed and implemented for the optimal route configuration. In the corresponding MADRL formulation, the drone-taxi related states, actions, and rewards are defined in this paper. Lastly, we confirm that our proposed algorithm achieves desired results.**

***Index Terms*—Smart City, Unmanned Aerial Vehicle (UAV), Drone-Taxi, Reinforcement Learning, Multi-Agent Deep Reinforcement Learning (MADRL)**

I. Introduction

Recently, the technology development in order to realize intelligent and autonomous smart city scenarios has been actively and widely discussed in various research societies, e.g,. electrical engineering, computer science, and civil engineering. Among various technologies, the use of unmanned aerial vehicle (UAV) networks is one of major interests in smart city applications [1]–[3]. The easily deployable and flexible natures of UAV networks facilitate the various emerging applications, e.g., automated and unmanned surveillance [4] and smart logistics such as drone-taxi [5].

In modern smart city logistics, unmanned autonomous aerial delivery is widely studied with the name of *drone-taxi*. For the drone-taxi applications, multiple drone-taxi UAVs are cooperated and coordinated in order to realize the efficient delivery of packages and passengers. Based on this requirement, multi-agent deep reinforcement learning (MADRL) algorithms are considered. Among various MADRL algorithms, we utilize a *G2ANet* algorithm which is for multi-agent cooperation [6]. Based on this *G2ANet*-based algorithm, multiple drone taxi agents make their own optimal paths to maximize their profits (i.e., maximize the amounts of passenger deliveries by multiple agents) in carpool systems. Our proposed *G2ANet*-based algorithm is evaluated with real-world scenarios and then we confirm that our proposed algorithm achieves desired performance improvements.

The rest of this paper is organized as follows. Sec. II introduces the basic concept of *G2ANet*; and Sec. III presents our proposed *G2ANet*-based algorithm for drone-taxi applications in smart city services. Sec. IV evaluates the performance of our proposed algorithm. Lastly, Sec. V concludes this paper and presents future research directions.

II. Reinforcement Learning Formulation: Graphic Neural Network with Attention Mechanism (G2ANet)

One of the main issues in *MADRL* is speculation about networking problems. In the case of the simplest form of *Deep Neural Network (DNN)*-based algorithm in previously existing *Deep Q-Network (DQN)*, the encoded input information that passes through the policy does not make a communicating connection. This result shows that the learning process proceeds while the agent does not cooperate with other agent and acts independently. For *Communication Network (CommNet)*, a new communication variable is created for every hidden layer. The new variables are taken and this information is directed to the next layer. Therefore new paradigm emerged in which *DQN*-based policy initiates communication with mutual agents [7]. However, in the case of *CommNet*, the weight of information shared between agents is all the same, thus there is a limitation on achieving high performance when individual agents are not equivalent to each other. An algorithm that can solve out the limit of *CommNet* is to use a graph structure and attention mechanism which is known as *G2ANet* [6]. In order to formulate the given problems with *G2ANet*, we need to define its corresponding graph and attention mechanisms.

A. Graph

Suppose there is a complex non-linear system. It is very difficult to try to represent the complex structure of this system in terms of variables and their equations. The best way to visually identify complex structures is to use graphs. The graph structure appears as a set of nodes and a set of edges that represent connecting information between nodes. The state information of all agents is mapped to each node. For each agent, the communication information of other agents is mapped to the edge. The set of edges is represented by the adjacency matrix. This series of processes is called abstraction of a complex environment, and is expressed as an encoding. The abstracted graph structure becomes the input of *hard attention* and *soft attention* of *G2ANet* policy, respectively.

B. Attention Mechanism

The attention method is mainly used in natural language processing (NLP) to find words that fit the context well. The

attention algorithm used in this paper is a hard attention and soft attention. The attention algorithm in this paper introduces the soft attention method, which combines hard attention and sequence to sequence (seq2seq) with the attention mechanism. For each encoded node, it collects information of all nodes other than itself, and puts it in the attention-layer. After that, we can get 1 or 0 by taking appropriate processing such as gumbel-softmax. A value of 1 indicates that the nodes are connected, and a value of 0 indicates that the nodes are disconnected. Then, we can obtain an adjacency matrix that shows important communicative relationship between all nodes. This is called hard attention. The soft attention method combines Seq2Seq. Each encoded node information is decoded into query, key, and value. Similar to the method presented earlier, for each encoded node, key and value pairs of all nodes other than itself are collected, and the scaled score is obtained by comparing it with its own query. This scaled score means a message to be delivered to all nodes except itself.

By masking the connection relationship obtained from hard attention, the node that will transmit information can be simplified and selective communication is possible. In other words, hard attention is a method to determine the connection relationship between nodes, and soft attention is a mechanism to determine which message to deliver between nodes. By combining these two attention methods, selective communication is possible. This is the core of *G2ANet*.

III. G2ANet-based Drone-Taxi Logistics

This section will explore the employment of drone-taxi UAVs in a smart city by using *MADRL*. Optimizing the path for a drone taxi platform and maximizing profit for the taxi industry is our considering scenario. The path planning works as the selection of an option to maximize profit by carpooling with several users or scheduling with a single user.

A. Scenario Description

Suppose there exists a number of drone taxi. Users can call on a drone taxi to get to their desired locations. Some users can even carpool, which will reduce the overall fees. Suppose there is a certain time it takes for a passenger to arrive at their destination. However, if a passenger carpools, the route may be wasted and may arrive later than the originally planned arrival time. Ideally, passengers pay less for the amount of time exceeding the optimal time of arrival. For the drone taxi to guarantee the maximum profit (*i.e.*, the optimal solution), the following considerations must be taken into account:

- Ensure to travel in the optimal path-planning route to carry as many users in one go.
- Ensure to minimize the path overlapped by other taxis.
- Select an option to maximize profit by carpooling with several users or scheduling with a single user.

B. Problems to use Reinforcement Learning

This scenario is more complicated than the previously mentioned surveillance drone situation. The problems are summarized below.

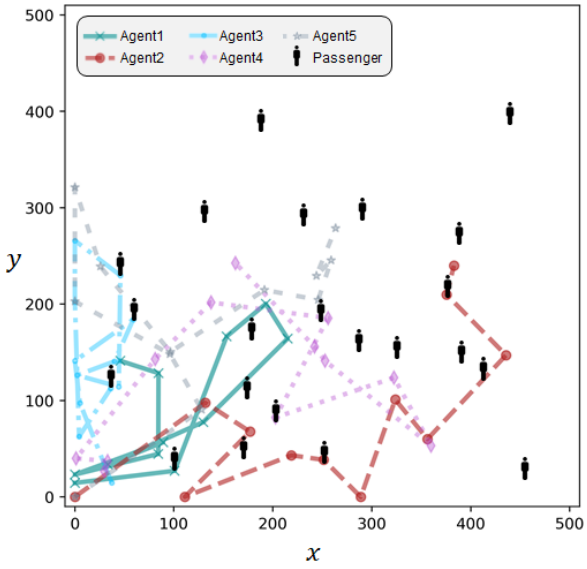

Fig. 1. Non-optimal oath-planning.

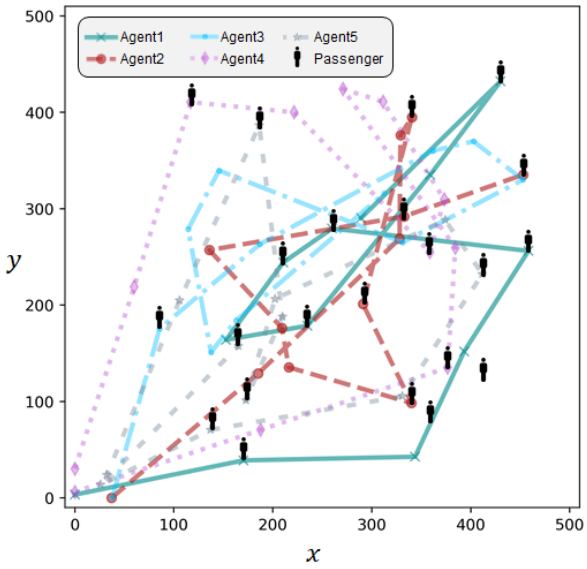

Fig. 2. Optimal path-planning using *G2ANet*.

- State: If the state of each drone taxi is predefined as its own location, relative location and distance information of other agents, user's relative location information, destination information, distance information, and number of customers it is carrying, there is a high possibility the policy may not be learned.
- Action: If the action is to move in the direction of 4, the carrying action would be heterogeneous. Thus, the problem would be how to carry the passenger. Let's suppose the other case. If the action is to carry all the passengers, then travel time is not considered, which is a problem since it does not satisfy Markov Property.

To resolve these problems, the process was altered to those mentioned below.

C. Candidate Passenger Selection

Drone taxis can only receive reservations from passenger within a certain range. In other words, they can partially observe the environment. Fig. **??** shows the rule of candidate selection. If the taxi had no customers, then the passenger who is closest to the taxi and satisfies both of the long-distance traveling criteria will be chosen as the candidate passenger. When there is more than one passenger in a taxi, current direction of the taxi and all the passenger's direction from which departure point to destination point in a vector form. Also, represent the distance from the candidate passenger's departure point to their destination point in a vector. Between these two vector of taxi and passengers, choose the one with the highest cosine similarity as the candidate passenger. Once there are no more seats available in the taxi, halt the process for choosing a candidate passenger.

D. Additional Environment Design

We consider four statuses for the passengers. First, the state that the passenger has not been assigned a taxi, second, the state that the taxi has been dispatched but did not take the taxi, third, the state that the passenger rides in a taxi, and the fourth, the state that get off the taxi. Only passengers in the first state (*i.e.*, not assigned a taxi) can be candidates. Also, there is a penalty for passengers on taxis. If the taxi takes more time than it took to go to the passenger's optimal route due to the carpool, the taxi has to bear the loss in the form of reducing costs for it.

E. States, Actions, and Rewards

For *MADRL* formulation, the designs can be carried out as follows. The drone taxi is the agent. The state configuration of the agent consists of the destination and penalty for the passenger. The total number of actions that the drone taxi can take was set to 4. This is to select the route for the destination point of passengers $1-4$.

Suppose a drone taxi (agent) selects one passenger as an action. Then, the drone taxi will move toward the passenger's destination point by the distance that can be moved during unit-time. If there is no passenger yet the drone taxi is set to select a passenger, it is designed to follow a random path. Finally, the rewards are designed as described below. For all passengers on the taxi, if the penalty is positive, a positive reward is given, and if the penalty is negative, a negative reward is compensated. In addition, moving the drone taxi for every unit-step taken will consume unnecessary energy, which will be given a negative reward.

IV. Performance Evaluation

In a 2D vector space of 500×500, five drone taxis are used as agents, and the number of passengers is always 20. When the carrying passenger gets off, a new passenger is created in a random location. The total episode time consists of 60 steps. Fig. 1 and Fig. 2 show the drone-taxi UAVs carrying passengers in the optimal route while carpooling. When the arbitrary action is given, it is confirmed that the agents are not able to properly pick up passengers and drop off at the destination, whereas when the action by *MADRL* is given, the agents reach the problem point and then move to the next problem point while following the optimal path.

V. Conclusions and Future Work

There are a lot of optimization algorithms, but especially in the environment that satisfies the Markov property, we introduce a method to optimize using reinforcement learning. We consider a scenario using multiple drones in a smart city, i.e., "how to maximize the common profit of drone taxis that can be carpooled". It is shown to optimize with *G2ANet*-based MADRL. We figure out the objectives to solve the scenario. The state design process, action design process, and reward shaping process to achieve the goal are shown. As a result, it is confirmed that the goal is achieved using *G2ANet*-based MADRL. When *G2ANet*-based MADRL is applied in the drone taxi scenario, it is confirmed that the total performance was better than the arbitrary decision-making actions.

As future research directions, surveillance applications are also worthy to consider, and then video streaming related efficient and effective algorithms can be considered [8]–[11].

Acknowledgment

This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funced by the Korea government (MSIT) (No. 2021-0-00467, Intelligent 6G Wireless Access System). Joongheon Kim is a corresponding author.

References

[1] M. Shin, J. Kim, and M. Levorato, "Auction-based charging scheduling with deep learning framework for multi-drone networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4235–4248, May 2019.

[2] S. Jung, W. J. Yun, M. Shin, J. Kim, and J.-H. Kim, "Orchestrated scheduling and multi-agent deep reinforcement learning for cloud-assisted multi-UAV charging systems," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5362–5377, June 2021.

[3] S. Park, W.-Y. Shin, M. Choi, and J. Kim, "Joint mobile charging and coverage-time extension for unmanned aerial vehicles," *IEEE Access*, vol. 9, pp. 94 053–94 063, 2021.

[4] D. Kim, S. Park, J. Kim, J. Y. Bang, and S. Jung, "Stabilized adaptive sampling control for reliable real-time learning-based surveillance systems," *Journal of Communications and Networks*, vol. 23, no. 2, pp. 129–137, 2021.

[5] W. J. Yun, S. Jung, J. Kim, and J.-H. Kim, "Distributed deep reinforcement learning for autonomous aerial eVTOL mobility in drone taxi applications," *ICT Express*, vol. 7, no. 1, pp. 1–4, 2021.

[6] Y. Liu, W. Wang, Y. Hu, J. Hao, X. Chen, and Y. Gao, "Multi-agent game abstraction via graph attention neural network," in *Proc. AAAI*, February 2020, pp. 7211–7218.

[7] S. Sukhbaatar, R. Fergus *et al.*, "Learning multiagent communication with backpropagation," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, December 2016, pp. 2244–2252.

[8] J. Kim, G. Caire, and A. F. Molisch, "Quality-aware streaming and scheduling for device-to-device video delivery," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2319–2331, August 2016.

[9] M. Choi, J. Kim, and J. Moon, "Wireless video caching and dynamic streaming under differentiated quality requirements," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1245–1257, June 2018.

[10] J. Koo, J. Yi, J. Kim, M. A. Hoque, and S. Choi, "Seamless dynamic adaptive streaming in LTE/Wi-Fi integrated network under smartphone resource constraints," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1647–1660, July 2019.
[11] M. Choi, A. F. Molisch, and J. Kim, "Joint distributed link scheduling and power allocation for content delivery in wireless caching networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 7810–7824, December 2020.