# Safety Enhanced Designs in UAS Risk Monitoring and Collision Resolution

by

Weichang Wang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved May 2021 by the
Graduate Supervisory Committee:

Lei Ying, Co-Chair
Yongming Liu, Co-Chair
Junshan Zhang
Yanchao Zhang

ARIZONA STATE UNIVERSITY

August 2021

# ABSTRACT

Collision-free path planning is also a major challenge in managing unmanned aerial vehicles (UAVs) fleets, especially in uncertain environments. The design of UAV routing policies using multi-agent reinforcement learning has been considered, and propose a Multi-resolution, Multi-agent, Mean-field reinforcement learning algorithm, named *3M-RL,* for flight planning, where multiple vehicles need to avoid collisions with each other while moving towards their destinations. In this system, each UAV makes decisions based on local observations, and does not communicate with other UAVs. The algorithm trains a routing policy using an Actor-Critic neural network with multi-resolution observations, including detailed local information and aggregated global information based on mean-field. The algorithm tackles the curse-of-dimensionality problem in multi-agent reinforcement learning and provides a scalable solution. The proposed algorithm is tested in different complex scenarios in both 2D and 3D space and the simulation results show that 3M-RL result in good routing policies. Also as a compliment, dynamic data communications between UAVs and a control center has also been studied, where the control center needs to monitor the safety state of each UAV in the system in real time, where the transition of risk level is simply considered as a Maikov process. Given limited communication bandwidth, it is impossible for the control center to communicate with all UAVs at the same time. A dynamic learning problem with limited communication bandwidth is also discussed in this paper where the objective is to minimize the total information entropy in real-time risk level tracking. The simulations also demonstrate that the algorithm outperforms policies such as a Round & Robin policy.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Prof. Lei Ying, for his invaluable supervision, continuous support and patience during the course of my PhD degree. Your immense knowledge, plentiful experience and insightful suggestions have encouraged me in all the time of my academic research and daily life.

I would like to thank Prof. Yongming Liu, whose expertise was invaluable in formulating the research questions and methodology, thank you for your patient support and for all of the opportunities I was given. Your guidance and feedback throughout this project brought my work to a higher level.

I would also like to thank Prof. Junshan Zhang and Prof Yanchao Zhang for their technical advice and support on my study.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

# Chapter 1

# INTRODUCTION

Unmanned aircraft systems (UAS) have been foreseen to play important roles in commercial, scientific and recreational applications, such as policing and surveillance, product deliveries, surveillance, air transportation Maza *et al.* (2009); Menouar *et al.* (2017). The number of UAVs in airspace has been increasing significantly over recent years. Because of that, flight planning has received increasing attention. This paper focuses on the flight planning where path planning, more specifically, is to identify a policy under which each UAV can maneuver over the airspace to reach their destinations. In particular, path planning in this paper is not to identify a fixed path, but refers to the process of dynamically and adaptively adjusting the path in real-time according to the surrounding environment following a policy, which is critical for operating UAVs in a dynamic environment such as uncertain weather conditions.

In this paper, we consider the collision avoidance problem in both 2D horizontal space and 3D space. We consider a system where all UAVs belong to the same entity so they will implement the same policy, which will be trained in a centralized fashion but would be executed decentralized. The state space or complexity of the policy increase exponentially as the number of UAVs increase, i.e. the curse of dimensionality. In this paper, we explore the mean-field approach for multi-agent reinforcement learning. The major difficulty in this multi-agent routing problem is how to develop polices under which each agent makes decision based on local information (and limited global information), but collectively reduce the risk of collision.

As a compliment, we also considers the problem of monitoring parallel Markov chains over wireless networks. The problem is motivated by risk monitoring in avia-

tion systems where a control tower needs to communicate with UAV in its region to monitor their risk levels. The solution of this problem can also be applied to other risk monitoring applications. The major challenge in the problem is that the communication bandwidth is limited. It has been shown in Galati *et al.* (2008), the channel becomes very congested when multiple UAVs in an area broadcast their positions through the ADS-B channel, which lead to significant data loss. This optimization problem is then formulated as a Multi-Armed Bandit (MAB) problem with the capacity of wireless channels as a hard constraint. The problem is similar to a restless bandit problem. The key difference is that the objective is to minimize the total information entropy of all bandits instead of finding the optimal bandit.

## 1.1 Related Work

Existing results on multi-agent UAV collision avoidance planning include both model-based methods and model-free methods. Model-based methods can provide near optimal performance, but require accurate information about movement models and environments, both of which are difficult to obtain in practice. With these information, some works model the collision avoidance as an optimal control problem, which could be solved by integer linear programming Richards and How (2002); Pallottino *et al.* (2002), nonlinear programming Raghunathan *et al.* (2004); Christodoulou and Kodaxakis (2006), pairwise optimization Carbone *et al.* (2006), gradient descent Zhao *et al.* (2020), random tree search Lin and Saripalli (2017) or many other methods Liu *et al.* (2020); Jenie *et al.* (2018); Yang *et al.* (2018). Data-driven reinforcement learning methods are model-free and have received increasing attention recently. For example, Chen *et al.* (2016); Lowe *et al.* (2017) proposed decentralized multi-agent reinforcement learning methods where each agent knows the positions and velocities of all other agents for taking actions. Such global information, however, is also difficult

to obtain in practice. As an alternative, Julian *et al.* (2019) proposed a centralized method based on neural networks for solving a pair-wise collision avoidance problem, and then a greedy method for a multi-agent system based on the parse-wise solution. There are also other approaches that solve the collision avoidance between UAVs and obstacles using Proximal Policy Optimization (PPO) Hu *et al.* (2020) and Long short-term memory (LSTM) network Singla *et al.* (2018).

In this paper, we consider solving collision avoidance path planning problem with a multi-agent decentralized partially observed reinforcement learning algorithm. Compare with centralized system, decentralized control systems do not have a central controller obtaining global information and making decisions for all agents in real time. Decentralized control is well used in large-scale complex systems, and serve as a effective tool to overcome specific difficulties arising in large scale complex systems such as high dimensionality, information structure constraints, uncertainty, and delays Bakule (2008). Decentralized control has even been used airspace system Šišlák *et al.* (2010). Partial observing is also a popular setting in decentralized system, the benefit of such a representation is that agents just need to plan only in terms of the small set of features, finding policy and value functions in low-dimensional spaces is typically easier and faster than finding value functions in high-dimensional global spaces. Some decentralized partially observable algorithms have good performance in robotic control Long *et al.* (2018) and even model based UAV control van den Berg *et al.* (2011).

Mean-field theory (MFT)Kadanoff (2009) studies models in which a large number of agents interact with each other. In MFT, the effect of all the other agents on a given agent is approximated by a single averaged effect, called mean-field, which reduces a many-body problem to a one-body problem. The main idea of MFT is to replace the interactions to the agent with an average or effective interaction.

Mean-field has been used in reinforcement learning recently Yang *et al.* (2018) for multiagent systems, called mean-field reinforcement learning. The method calculates the mean actions of other agents based on the neighboring agents, and uses it as the input of a single-agent reinforcement learning algorithm. Therefore, the size of the state space for the single-agent reinforcement learning problem remains a constant even as the total number of agents increases. While this mean-field reinforcement learning approach provides a scalable solution, it cannot be directly applied to the routing problem as it is critical for each UAV to maintain a safe distance to other UAVs, so the any action based on the aggregated "mean" information is not sufficient for safety.

## 1.2 Main Results

In this paper, we consider the routing problem where UAVs need to reach their destinations while maintaining safe distances from each other. The UAVs (also called agents in this paper) make their own decisions based on their own states and observations. To overcome the curse of dimensionality and maintain safe distances, we implement a mean-field reinforcement learning approach with multi-level information. Each agent has a local view space which it has accurate information of its neighbors in the space (called intruders) such as the relative position, velocity and its destination direction, and has statistical information about other agents outside of the viewspace. Based on that, we proposed a multi-resolution, multi-agent, mean-field reinforcement-learning algorithm to solve the collision avoidance problem where each agent only uses local observations and aggregated mean-field for decision making, and the global information statistics are used for training the reinforcement learning algorithm. In particular, we use the actor-critic algorithm with convolutional neural network (CNN).

The paper is organized as follows. First, we consider about the collision avoidance reinforcement learning algorithm in 2. In this chapter, the problem formulation of the proposed collision avoidance model based on probabilistic dynamic is introduced. Then the methodology of the proposed mean-field reinforcement learning algorithm is explained. The simulation results are presented to show the convergence of the learning process. Comparisons between the proposed method and some others based on cost and average distance are also provided. To make the simulation concrete, we also build an environment on MATLAB&SIMULINK UAVToolBox Inc (2020). Further more, we have extended our algorithm into 3D scenario, simulation results have also been provided. Further analysis is discussed and several conclusions are drawn. Then, we consider about the monitoring of central controller under limited bandwidth. We prove that the problem is indexable for single-rate wireless channels and establish a sufficient condition under which the problem is indexable with multi-rate wireless channels. Our numerical evaluations show that our algorithm outperform other heuristics such as the greedy policy and Round& Robin policies.

Chapter 2

# *3M-RL:* MULTI-RESOLUTION, MULTI-AGENT, MEAN-FIELD REINFORCEMENT LEARNING FOR AUTONOMOUS UAV ROUTING

## 2.1 Problem Formulation

We consider the horizontal routing (also called flight planning), i.e. paths over a two-dimensional space. We discretize the two dimensional space into in a grid, and let $s_i \in \mathcal{S}$ denote the relative position of the $i$th agent on the grid relative to its destination in an $L \times L$ space. Besides its own state, an observation $o_i \in \mathcal{O}$ on about its viewspace is also available at each time slot, as shown in Fig.2.1. The viewspace consists of two parts, the first part is area close to the agent, which requires detailed information of "intruders" in the view space such as their relative positions, heading directions and distance from the destinations. The second part is a mean-field viewspace which represents the area surround detail viewspace, which does not include detailed information, only number of intruders and their mean direction in each meta-grid are required. The number of agents together with the mean direction represent the "flow" of intruders in that area. Number of intruders characterized the size of flow, and the mean direction represents flow's direction. The detailed space is used to avoid the intruders close to it, and the mean viewspace helps the agents to avoid the congested area.

We define $r_i(s_i, o_i)$ to be the reward of the $i$th UAV at state $s_i$, and with obser-

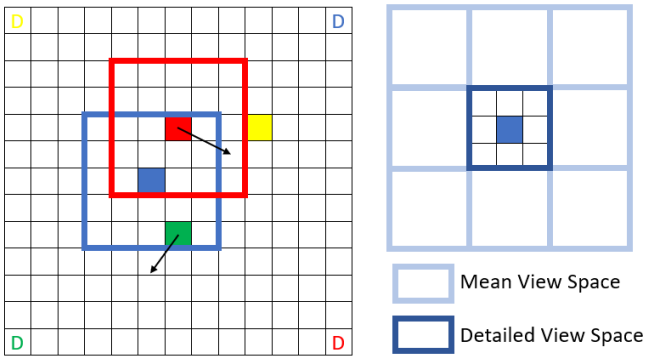



**Figure 2.1:** The Grid Space and Viewspace of Agents

vation $o_i$:

$$r_i(s_i, o_i) = \tag{2.1}$$

$$\begin{cases} -c_0 \text{dist}(s_i, d_i) - c_1 \sum_{j \in \mathcal{N}(i)} \exp(-\text{dist}(s_i, s_j)) - c_2 \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{when } s_i \neq d_i \\ r \qquad \text{when } s_i = d_i \end{cases}$$

where $d_i$ is the destination of the $i$th UAV, $\text{dist}(\cdot, \cdot)$ is the Euclidean distance function, $\mathcal{N}(i)$ is the set of UAVs in the viewspace of the $i$th UAV. $c_0, c_1, c_2, r$ are positive constants. When the UAV arrives at its destination, a positive reward $r$ is given. Otherwise, a cost (negative reward) is incurred. The cost is consists of three components. The first component $-c_0 \text{dist}(s_i, d_i)$ encourages the UAVs to go towards their destinations following shortest paths if no collision is detected. The second component $-c_1 \sum_{j \in \mathcal{N}(i)} \exp(-\text{dist}(s_i, s_j))$ encourages the UAVs to keep distance with each other for safety. An intruder close to an agent incurs a large cost due to safety concerns. More discussion will be given in the learning process. An intruders far away from an agent results in a small cost. The intruders outside the view space do not change the current cost, however, they are included in the training as the mean-field. By doing this, multiresolution is realized in the training process. The last term $-c_2$ is

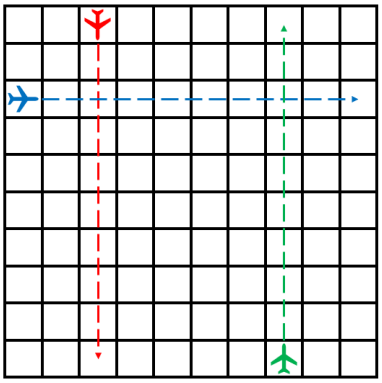


**Figure 2.2:** Without the Constant Term $c_2$, There Might Be a Collision

a constant and is used to balance the first term and the second term. This constant term plays a significant roll in the model. We next present a toy example for illustrating its importance. As shown in Fig.2.2, there are three UAVs in a $10 \times 10$ grid space, where dashed arrows are their direct path toward the destinations. Let $c_0 = 1$ and $c_2 = 0$. The view space is a $5 \times 5$ area around the agent. To avoid the collision between the blue and red UAV, it would be a good choice for the red one to take a "left" move at its first step, or equivalently, the blue one to take an "up" move. To encourage one of the agents in changing its trajectory, its total cost has to be smaller than the shortest path. However, in the first step, the first term dominates the cost function, and the first term leads to a large cost at least 9 in the final total cost. As a result, to actuate the movements, $c_1$ has to be greater than 5.6094. However, $c_1$ is designed to be a parameter to control the safe distance between the agents, so a larger $c_1$ makes the blue one keep far away from the green one, since the second term dominates the cost at this time, and a bad trajectory will be trained. This problem becomes worse when the size of the grid space increase. As a result, $c_1$ has to be large enough to achieve the collision avoidance, the first term should dominate in $r_i$ anytime there is a collision. Based on this, $c_2$ has to dominate in $r_i$ in collision free states. We know that all direct paths towards destinations take the same number of

steps in the grid space, $c_2$ term is not able to select the path close to the shortest path, $c_1$ is used to adapt the shortest path, it should be smaller than $c_2$ in general. The simulation comparison is shown in Section 2.3. Constants $c_0, c_1, c_2, r$ need to be tuned during the training.

In this problem, we need optimizes its policy $\pi(s_i, o_i) : \mathcal{S} \times \mathcal{O} \to [0, 1]$ to minimize the discounted total cost $\sum_t \gamma^t r_i \left(s_i(t), o_i(t)\right)$, where $\gamma$ is the discount factor.

## 2.2 3M-RL for UAV Routing

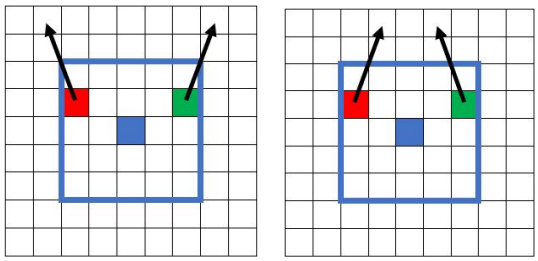
To solve the problem, we propose a multi-agent reinforcement learning approach called 3M-RL. This is a model free method, and as we mentioned in Section **??**, the policy is optimized by a well designed reinforcement learning process. 3M-RL is also a multi-agent system, where every agent makes its own decision independently. As a result, the complexity can be significantly reduced compared with methods like dynamic programming which considers all agents together. In this section, we will first describe a method called Mean-Field Multi-agent Reinforcement Learning, and discuss about its issue when applying to collision avoidance in Section 2.2.1. Then details about our 3M-RL approach and implementation will be discussed in Section 2.2.2 and 2.2.3. We will start from 2D horizontal space domain, and then extend our algorithm into 3D domain in Section 2.2.5.

### *2.2.1 Mean-Field Multi-agent Reinforcement Learning*

Mean-Field Multi-agent Reinforcement Learning is proposed in Yang *et al.* (2018), where the key idea is the following equation:

$$Q^j(s, \mathbf{a}) = \frac{1}{N^j} \sum_{k \in \mathcal{N}(j)} Q^j(s, a^j, a^k) \tag{2.2}$$

$$\approx Q^j(s, a^j, \bar{a}^j) \tag{2.3}$$



(a) Case 1 (b) Case 2

**Figure 2.3:** A Toy Example about the Error in Approximation

where $Q^j$ is the Q value of agent $j$, $\mathbf{a}$ is the joint action of all the agents, $\mathcal{N}(j)$ is the index set of the neighboring agents of agent $j$ with size $|\mathcal{N}(j)| = N^j$, and $\bar{a}^j$ is the mean action of $j$'s neighbor. However, in our collision avoidance flight planning problem, only an approximation like Eq.2.3 might lead to some error. It can be shown by a simple example, as in Fig 2.3. Consider the blue vehicle as agent $j$, and we focus on its Q values $Q^j(s, \mathbf{a})$. The blue frame is its view space, containing the red and green vehicles as its neighbors or intruders in other words. The arrow on the red and green vehicles are their actions, and their mean action forms $\overline{a}_j$. Consider these two cases in Fig.2.3(a) and Fig.2.3(b), the only difference between them is that the actions of the red and blue agents are switched. As a result, the mean action $\bar{a}^j$ in these two cases are exactly the same. Assume the blue agent's action $a^j$ is going up, it is obviously that case 2 in Fig.2.3(b) has a high risk of collision and a large cost, but the case 1 in Fig.2.3(a) is collision free. Therefore, the Q value of the blue agent $Q^j(s, \mathbf{a})$ cannot be simply approximated as $Q^j(s, a^j, \bar{a}^j)$.

In our problem, similar to Yang *et al.* (2018), agents focus on the intruders close to it, as shown in Fig.2.1. Other than the mean observation over its neighborhood area, each UAV focuses on the intruders in its view space. The features like heading direction, destination of intruders in the view space, detailed actions are used by its neighborhood $\bar{a}^j$. These information are considered as observation $o_j$. The mean

viewspace is also necessary. An agent may not able to avoid intruders in a congested viewspace, so mean direction is used because detailed information is not necessary for congestion avoidance purpose.

### *2.2.2 3M-RL: Multi-Resolution, Multi-Agent, Mean-Field Actor Critic Algorithm*

As we mentioned in Section 2.1, other than the mean field observations, we add detailed viewspace to solve the problem. We implemented the algorithm with CNN and Actor-Critic, where CNN is used to extract features from observations and the actor-critic method is used to reduce the variance of learning. Actor-critic method is one of the reinforcement learning implementation that have a separate memory structure to explicitly represent the policy independent of the value function, and require minimal computation in order to select actionsSutton and Barto (2018). It is also well used in many multi-agent games, and many of them have very good results like Lowe *et al.* (2017). The combination of CNN and actor critic is also popular Pinto *et al.* (2017); Christodoulou (2019), since CNN is a good structure to deal with image-like input.

First recall the policy gradient method Sutton and Barto (2018):

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\pi}\left[\sum_{a} q_{\pi}(s_i, a_i) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t)\right] \tag{2.4}$$

$$= \mathbb{E}_{\pi}\left[G_t \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t)\right]. \tag{2.5}$$

where $\pi(a_t|s_t) : \mathcal{S} \times \mathcal{A} \to (0, 1)$ is the policy, and it means the probability of taking the action $a_t$ at state $s_t$, this policy function is approximated with parameter vector $\boldsymbol{\theta}$. If the parameters are updated as Monte-Carlo Policy Gradient as Eq.(2.5), it has a large variance because the cumulative return $G_t$ has a large variance, and the log probability $\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t)$ is large. So the learning converges slowly. The policy gradient theorem Eq.(2.4) can be generalized to include a baseline policy $b(s)$ to

reduce the variance:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) &= \mathbb{E}_{\pi}\left[\sum_{a} q_{\pi}(s_i, a_i)\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t)\right] \\ &= \mathbb{E}_{\pi}\left[\sum_{a} \big(q_{\pi}(s_i, a_i) - b(s)\big)\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(a_t|s_t)\right]. \end{aligned} \tag{2.6}$$

Since $\sum_a \big(q_\pi(s_i, a_i) - b(s)\big) = 0$, the new implementation with the baseline is unbiased. By selecting appropriate baseline value, large return $G_t$ leads to large $b(s)$, the variance is reduced, which is a well-known result, called variance reduction. In the actor-critic method, another system $\hat{v}(s_i, \mathbf{w})$ is used to learn the state value, where $\mathbf{w}$ is the parameter.

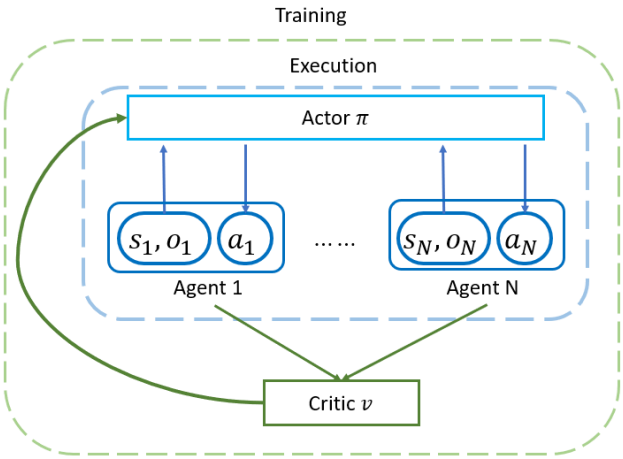



**Figure 2.4:** Overview of Mean Field Actor Critic Method

Based on the discussion above, the overview of our 3M-RL is shown in Fig.2.4. The information about mean actions of neighbors as we discussed before, is contained in the observation $o_i$. Although all the agents make decisions on their own state and observation, they can also cooperate with each other since all the agents in the system follow the same policy $\pi_\theta(a_i|s_i, o_i)$, by assuming that the intruders follow the same policy as itself. Here the policy $\pi$ is parameterized by $\boldsymbol{\theta}$, and it is independent with agents' index $i$. Unlike many multi-agent reinforcement learning algorithms that all

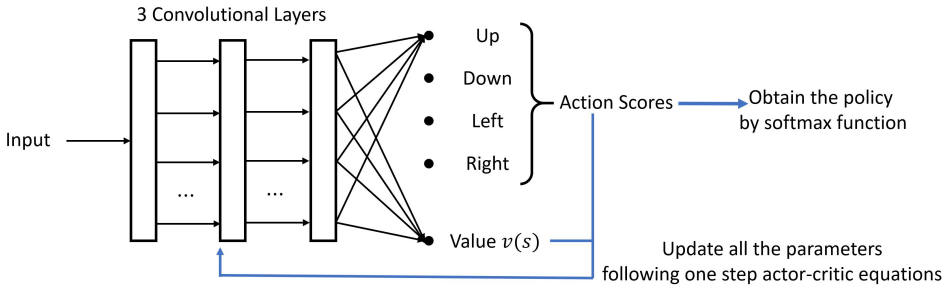



**Figure 2.5:** Structure 1: Actor Critic Network

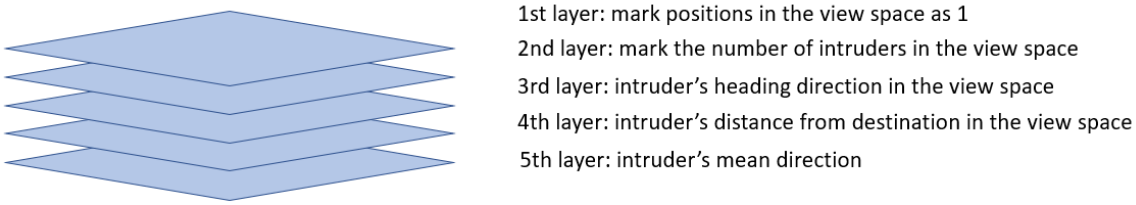



**Figure 2.6:** Input Matrix of Structure 1

agents has its own policy, homogeneous agents setting here simplify the complexity of the policy, and also help with the cooperation between agents. A value estimator will evaluate the value functions of each agent, and updates the policy $\pi$. We also defined the critic network $\hat{v}(s_i, o_i)$ be the estimator parameterized by $\mathbf{w}$. The estimator and policy functions are approximated by neural networks with parameters $\boldsymbol{\theta}$ and $\mathbf{w}$.

The algorithm is presented in Alg.1. Line 3 to 6 is the testing part, where $T$ is the time horizon of the episode, and $t$ is the time index in the episode. In this part, all agents act following the current policy $\pi$. The sample paths (simulation data) are recorded. Line 8 to 13 is the policy updating part, where the algorithm updates two sets of parameters, $\boldsymbol{\theta}$ which is the set of actor parameters and determines the policy, and $\mathbf{w}$ wich is the set of critic parameters and estimates the state value. The policy of an agent depends on the action scores and follows the Boltzmann distribution:

$$\pi(a_j) = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{2.7}$$

where $z_j$ is the score of the $j$th action in action space $\mathcal{A}$.

We would like to remark that a single trace might introduce large variance, so it

**Algorithm 1:** 3M-RL: Multi-Resolution Multi-agent Mean-Field Actor Critic Reinforcement Learning

**Input:** A policy parameterization $\pi(a|s_i, o_i, \boldsymbol{\theta})$ and a state value parameterization $\hat{v}(s_i, o_i, \mathbf{w})$

**foreach** *Episode* **do**

  **while** *Not Arrived and* $t<T$ **do**

    Take action $a(t) \sim \pi(\cdot|s_i(t), o_i(t), \boldsymbol{\theta})$;

    Get next relative position $s_i(t+1)$, next observation $o_i(t+1)$, and reward $r_i(t)$;

    Record $\big(s_i(t), o_i(t), \pi(\cdot|s_i(t), o_i(t), \boldsymbol{\theta}), r_i(t)\big)$

  **end**

  **foreach** $s_i(t)$ *in the episode* **do**

    $G_t \leftarrow \sum_{\tau=t}^{T} \gamma^{\tau} r_i(\tau)$;

    $\delta_t \leftarrow G_t - \hat{v}(s_i(t), o_i(t), \mathbf{w})$;

    Network parameters update: $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta_t \nabla \hat{v}(s_i, o_i, \mathbf{w})$;

    Network parameters update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} \delta_t \nabla \ln \pi(a(t)|s_i(t), o_i(t), \boldsymbol{\theta})$

  **end**

**end**

is better to use batch update. We assume that the number of traces we collect in an episode fits for the batch size, so the policy can be updated at the end of each episode. Otherwise when there are too many agents in the system, the policy need to be updated during an episode. In this case, the data in line 6 need to be substituted by the following two steps:

1. $G_t(s_i, o_i) = r_i(t) + \gamma \hat{v}(s_i(t+1), o_i(t+1), \mathbf{w})$

2. Record $\big(s_i(t), o_i(t), \pi(\cdot|s_i(t), o_i(t), \boldsymbol{\theta}), G_t(s_i(t), o_i(t))\big)$

where the $G_t(s_i, o_i)$ is estimated return which is used as $G_t$ directly in line 9, since the future rewards might not saved in the same batch.

It is worth noticing that Alg.1 is just a training process, the policy determined by the actor is indeed a stochastic policy, because stochastic exploration is necessary during the training process. However, after the policy is well trained, the policy we used to execute is a greedy policy, choosing the action with highest action score, then it becomes a deterministic policy and exploration is not required any more.

### *2.2.3 CNN Implementation*

We use CNN in our actor-critic algorithm because the state and observations of an agent can be easily transferred to image. We will use CNN to approximate the $\pi(s_i, o_i, \boldsymbol{\theta})$ and $\hat{v}(s_i, o_i, \mathbf{w})$ in Alg.1.

We propose two CNN structures for this algorithm. The first structure is shown in Fig.2.5, which is a three-layer CNN used to extract features from the input matrix, and whose output consists of actor part (action scores) and critic part (state value). The actions of the agents depend on the corresponding action scores and Eq.2.7. The input of the system is a five layer matrix shown in Fig.2.6. All the information such as relative positions and observations can be described by an image. For example,

for the blue and red agents in the toy example Fig.2.2, the input of the agents can be shown as Fig.2.7. We assume that the square at the center is the destination, and for the blue agent, all its nonzero entries settle in the blue square, these entries characterized its observation at current time. The state information is illustrated by the position of the square, and the state is the relative position of the center of the blue square relative to the destination. As a result, the agent in a $L \times L$ space has input matrix with size $(2L-1) \times (2L-1) \times 4$, and it contains all the state and observation information. This structure fits for the case when the dimension of view space is small, and the features can be extract from the viewspace is very limited, it would be better for us to regard the whole space as an image, and use it as the input of CNN.

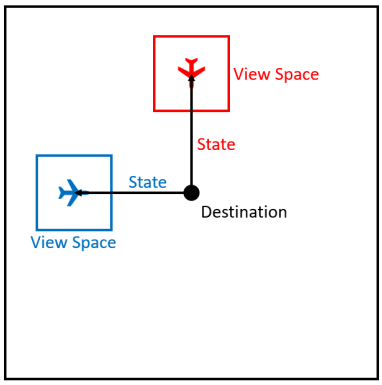



**Figure 2.7:** Input Matrix in Horizontal 2D Space

The above structure turns out to be inefficient when the grid space is large. Since only a few entries have nonzero values. Based on this, we propose the second CNN structure, as shown in Fig.2.8, which is a Mixture Feature Embedding neural network. CNN is used to extract the observation features in viewspace, and the state feature is just binary code of the state. A concatenate layer is used to concatenate these features, and the output have the same type as in Fig.2.5. In this structure, the input

of the CNN is not the matrix in Fig.2.6. Instead, it is a three-layer matrix with size equals to the viewspace. The entries in the matrix is the number of intruders in the view space, intruder's heading direction, and intruder's distance from destination, just as the second, third, and fourth layers in Fig. 2.6. This structure significantly reduces the processing time of the neural network as size of grid space increases.

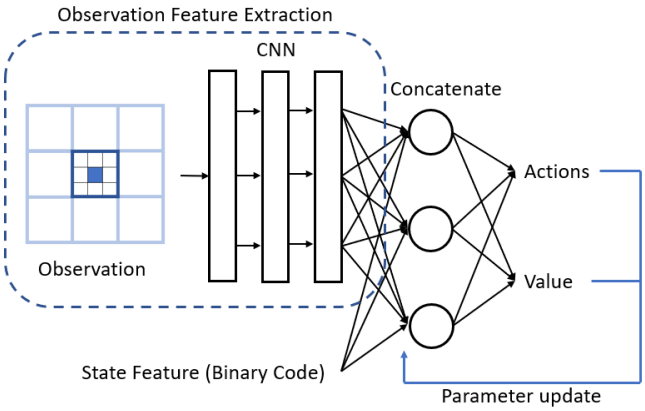



**Figure 2.8:** Structure 2: Mixture Feature Embedding Neural Network

### *2.2.4 Continuous State Space Extension*

The collision avoidance algorithm may be difficult to implement in some real problems, we also extended the algorithm to continuous state space. The continuous state space environment is built on MATLAB SIMULINK with UAV ToolBoxInc (2020).

In this environment, the positions are no longer integers, it could be any point in the space, the action is also not simple as before, it is adapted to a turn set: $\mathcal{A} = \{Left - Left, Left, Maintain, Right, Right - Right\}$, which is achieved by setting the target roll angle to some predefined values.

It is worth mentioning that the information in the state and observations also need to be extended. Structure 2 in Fig.2.8 is used, and we include the agent's heading direction in the state feature vector, and the 3rd layer in the observation matrix in

Fig.2.6 is also adapted to continuous heading direction. To prevent frequent sharp turning, we added a small cost to $r_i$ every time when turning action is made. Details of the simulation and results are shown in Section 2.3.

### *2.2.5 Continuous 3D State Space Extension*

We've also considered solving collision avoidance path planning problem in 3D space with our 3M-RL algorithm. Collision avoidance in 3D space can be much more complex than horizontal 2D scenario, since it extends the state space. On the other hand, different from collision avoidance in 2D space, UAVs have different model equations in horizontal and vertical directions Inc (2020), as a result, they may have different turning radius and sensitivity in different directions. In general, the target of collision avoidance path planning is no intruders in its collision hull. Collision hull is the risk area which center coincides with the UAV's position, it can be set as cylinder Ferrera *et al.* (2018); Zhu *et al.* (2016) or simply a sphere Lin *et al.* (2020). Fig.2.9 illustrates an example of cylinder collision hull, which will also be used is our simulation. For generality, we defined a reward function in 3-D space as below:

$$r_i(s_i, o_i, a_i) = \tag{2.8}$$
$$\begin{cases} -c_0 \text{dist}(s_i, d_i) - c_1 \sum_{j \in \mathcal{N}(i)} \mathbb{1}\{s_j \text{ is in collision hull}\} \\ \qquad\qquad -c_2 - r(a_i) \qquad \text{when } s_i \neq d_i \\ r \qquad \text{when } s_i = d_i \end{cases}$$

where *dist* function here is the distance in 3-D space, and $\mathbb{1}$ is the indicate function. We also introduced a reward component on action $r(a_i)$ that is used to control the preference of actions, because in some applications, the raising of UAVs may leads to more energy costs. We have also set up simulations illustrating the utility of this component.

In this part, we still consider continuous state space, but the environment is no longer built on SIMULINK UAV ToolBox, because in this ToolBox, the height of fixedwing UAV model is controlled by setting target height, which is much slower than the control on row angle. To simplify the model, we defined the action space of the UAVs as:

$$\mathcal{A} := \{Left - Left, Left, Maintain, \\ Right, Right - Right, Up, Down\} \tag{2.9}$$

and details shown as in Tab.2.1. In Tab.2.1, $HA$ is the variation in heading angle, and $RD$ is the raising or descending in height, we also assume that these actions are finished fast enough compare with one time slot. This is just a preliminary model, the real actions largely depends on the control system and model of UAV, we set up this model to test the ability of our algorithm handling states and observations in 3-D space.

CNN has also been well used in many 3-D object recognition and classification of 3-D images Zhang *et al.* (2018); Maturana and Scherer (2015). We use the same structure as in Fig.2.8, simply substitute the layers in CNN as 3D CNN layers. And it should be noticed that the size of action is 7 as we mentioned in Eq.2.9. The detailed view space and mean view space in the observation is illustrated in Fig.2.10, and the input of the 3D CNN is a 4D matrix (width $\times$ length $\times$ height $\times$ information length), where the information length we used is 5, just as the layers in Fig.2.6. On the other hand, the algorithm and the equations we used to update neural network parameters are still as shown in Alg.1.

| Actions | LL | L | M | R | RR | U | D |
|---|---|---|---|---|---|---|---|
| HA | $-30°$ | $-15°$ | $0°$ | $15°$ | $30°$ | $0°$ | $0°$ |
| RD | 0 | 0 | 0 | 0 | 0 | $+1$ | $-1$ |

**Table 2.1:** Details of Actions in 3D Space

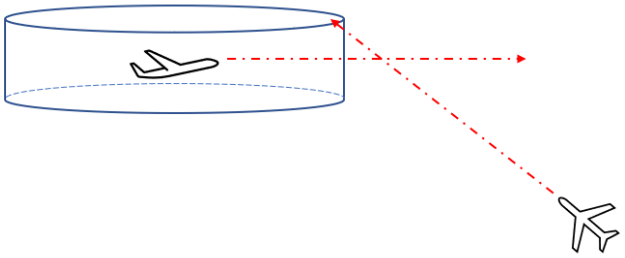


**Figure 2.9:** Collision Hull in 3D Space

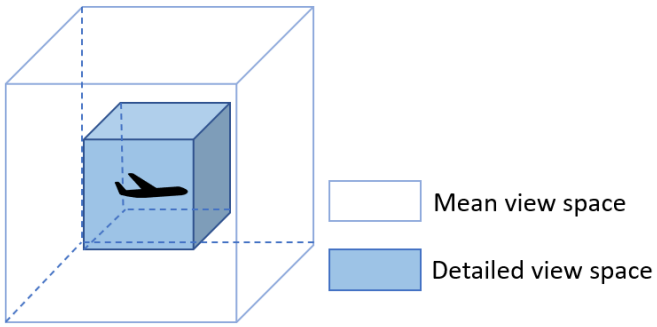



**Figure 2.10:** Observation in 3D Domain

## 2.3 Experimental Results

We evaluated 3M-RL on 10×10, 20×20, and 50×50 grid respectively, and on a continuous environment built on UAVToolBox in MATLAB&SIMULINK. We remark that in this section, to have a clear estimation on the plotted trajectories, the policy used to test is the greedy policy instead of following the Boltzmann distribution Eq.2.7, so the testing policy is deterministic.

### 2.3.1 *Experimental Results on the* $10 \times 10$ *Grid Space*

We first considered a system where four agents move in a $10 \times 10$ grid with $5 \times 5$ detailed viewspace around it. Each agent starts at one corner of the space and heads toward the cross corner. We chose this moderate-size network so we can solve the minimum cost by using dynamic programming as the baseline to evaluate the performance of the proposed algorithm. Since the view space it too small in this part, so neural network with structure 1 is used.

We first show the comparison in convergence in Fig.2.11, it plots the total cost of all vehicles in the training process. Our proposed 3M-RL is compared with two other algorithms. Mean Field Q is also a multi-agent reinforcement learning. It is implemented by regrading the input of the CNN as the state, and training the policy use tabular Q-learning. Because of the large state space, it takes a very long time to converge. As we can see, after 10,000 episodes' training process, the decrease tendency of mean-field Q-learning is not clear, and the proposed 3M-RL used Actor-Critic structure and converges much faster. This fast convergence is the reason we chose to use the Actor-Critic algorithm.

We also compared the proposed algorithm with the mean-field multi-agent reinforcement learning algorithm proposed in Yang *et al.* (2018). As we pointed out in the earlier section that the direct application of the mean-field multi-agent reinforcement can lead to unsafe plans, because of the lack of significant neighbor information. Fig.2.11 shows the total costs under our method and pure mean-field method Yang *et al.* (2018) in training process. We can clearly see that 3M-RL results in a much lower cost and is close to the optimal.

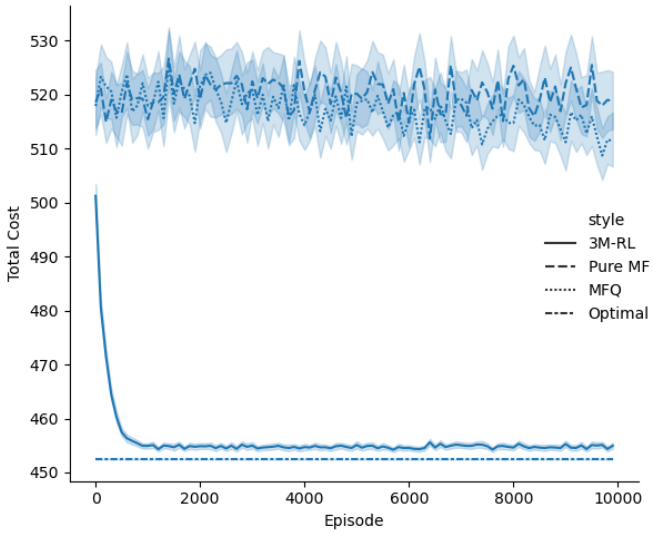



**Figure 2.11:** Convergence Comparison

### *2.3.2 Experimental Results on the* $20 \times 20$ *Grid*

We considered four agents on a $20 \times 20$ grid, where each agent has $11 \times 11$ detailed viewspace around it. Also in this section, we use CNN with structure 1. First our experiment showed the significant of the constant term $-c_2$ in the reward function Eq.(2.1). As shown in Fig.2.12 and Fig.2.13, the UAVs have same start point and destinations in these two cases, however, $c_2$ in Fig.2.12 balances the first and second term in the cost function Eq.2.1 well (we consider $c_0 : c_1 : c_2 = 0.05 : 5 : 1$). As we can see in the figure the collision is well solved. Fig.2.13 have parameters $c_0 : c_1 : c_2 = 1 : 5 : 0$, without the $c_2$ term balancing the reward function, a collision occurred.

To show the advantage of using CNN, we also design another actor-critic feedforward neural network with 2 hidden layers. The input of this network is the binary coded information of the state and observation of the UAV. Fig.2.14 shows the average distance between all agents in a episode. The horizontal axis represents the time in an episode, and the vertical axis represents the average distance between each pair

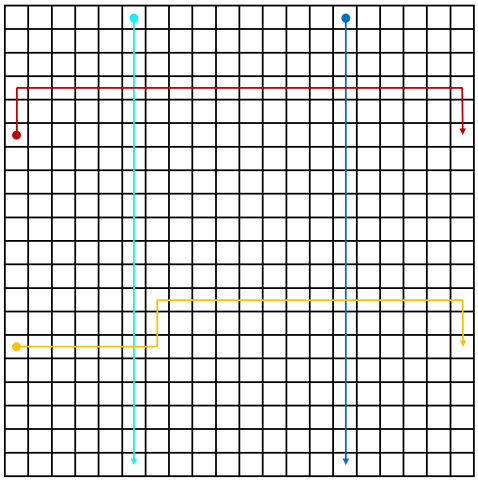


**Figure 2.12:** Trajectory Planning with $c_2$

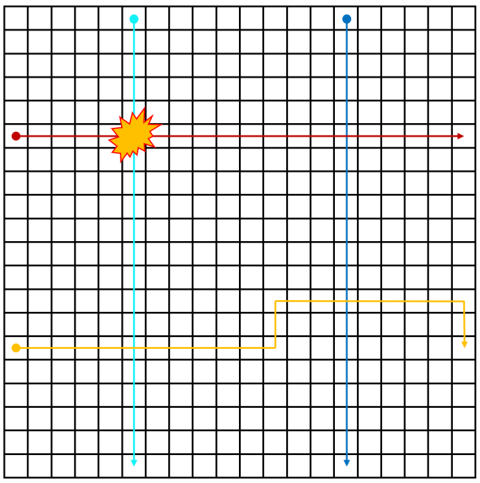


**Figure 2.13:** Trajectory Planning without $c_2$

of UAVs. At time 0, all UAVs start from their origins, so the distance at this time is large. As they approaches their destinations, they first get closer to each other, which increases the risk of collision. When the UAVs are close to their destinations, the pairwise distance again becomes large. As a result, larger average distance represents a better path planning policy. As shown in Fig.2.14, the proposed mean field actor critic method with CNN has the best performance.

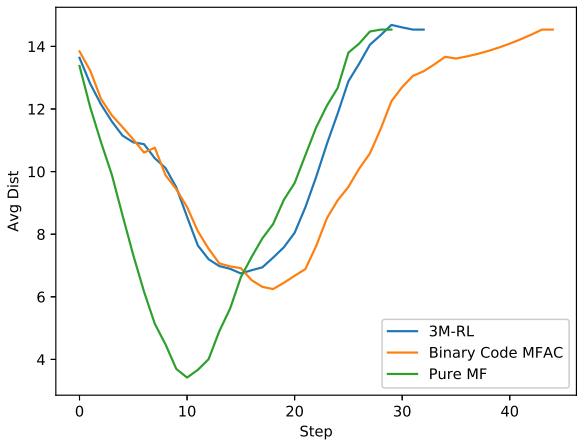



**Figure 2.14:** Average Distance Comparison

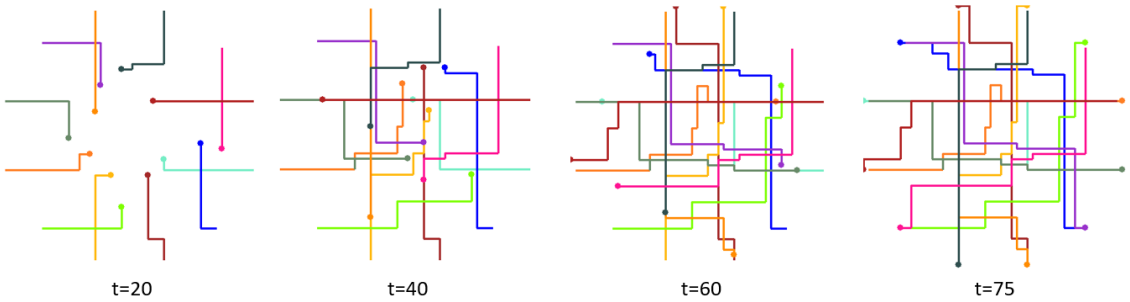



**Figure 2.15:** 3M-RL Simulation Trajectories of 12 UAVs in $50 \times 50$ Grid Space

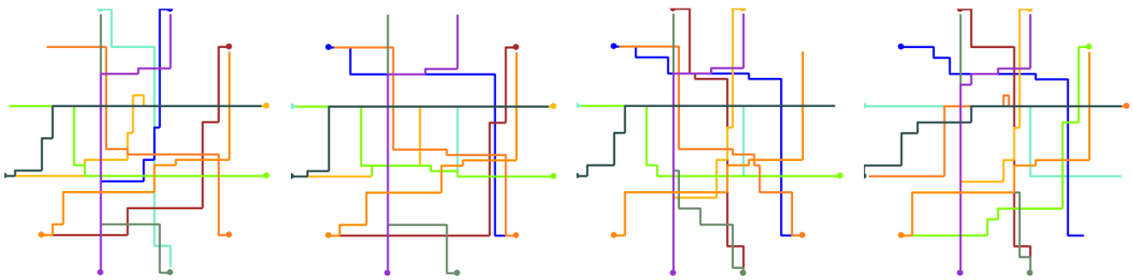


**Figure 2.16:** Remove Two of the UAVs, Trajectories of the Remaining UAVs

### *2.3.3 Experimental Results on a* $50 \times 50$ *Grid*

The trajectory plot of 12 UAVs in $50 \times 50$ grid space is shown in Fig.2.15. In this simulation, each agent has a $21 \times 21$ view space around it, and $c_0 : c_1 : c_2 = 0.1 : 10 : 1$. Since the grid space is large, CNN with structure 2 was used. We plotted the

trajectories they have passed as the time index $t = 20, 40, 60, 75$. We can see that none of the agents collide with others.

To show the importance of mean viewspace, we also evaluated the our algorithm (v1) and the case when the mean direction layer in Fig.2.6 is removed (v2). We compared the minimum distance to the others and the maximum time slot cost of each agent during an episode. It shows that with the mean direction layer, the agents have better safety distance. Without the 5th layer in the input matrix, the maximum cost suffer from larger variance, which means there is some congestion during the episode.

To test the robustness of the algorithm, we also tested the policy when 2 of the 12 UAVs are removed, each UAV will suffer from less intruders. The results turned out that most trajectories remains the same.

### *2.3.4 Continuous Space Simulations*

As we mentioned, our continuous space simulation is built on UAVToolBox in SIMULINK Inc (2020).

The environment is set to be discrete in time, continuous in state space and discrete in action space. The action is made every second, and the action is some predefined target roll angle. We used the fixed wing UAVs as our agents. To simplify the problem, the speed of UAVs remains the same the all the time, and the pitch angle is fixed to 0. We also would like to mention that after the target roll angle set, the variation of roll angle follows the step response defined in the UAVToolBox, as shown in Fig.2.17. This makes the mechanism of environment more complicated. Here the reward parameters we used is $c_0 : c_1 : c_2 = 0.1 : 10 : 1$. To prevent frequent sharp turning, we also added a small penalty on the turning actions. We simulate the collision avoidance problem of 4 UAVs, and the trained trajectory is shown in

| | Min Dist | | Max Cost | |
|---|---|---|---|---|
| Vehicle | v1 | v2 | v1 | v2 |
| 1 | 2.83 | 1.00 | 2.01 | 11.30 |
| 2 | 2.83 | 2.83 | 2.05 | 1.99 |
| 3 | 2.83 | 2.24 | 2.02 | 3.43 |
| 4 | 2.83 | 1.00 | 1.79 | 11.50 |
| 5 | 2.83 | 2.83 | 1.77 | 1.77 |
| 6 | 2.83 | 2.83 | 2.31 | 1.82 |
| 7 | 2.83 | 2.83 | 2.21 | 1.96 |
| 8 | 2.83 | 2.83 | 1.78 | 1.85 |
| 9 | 2.83 | 2.24 | 1.96 | 3.31 |
| 10 | 2.83 | 2.24 | 1.78 | 3.59 |
| 11 | 2.83 | 2.83 | 1.78 | 1.78 |
| 12 | 2.83 | 2.83 | 1.78 | 2.15 |

**Table 2.2:** Comparison of the Input Matrix with Mean Direction Layer (v1) and Without Mean Direction Layer (v2) in Fig.2.6

Fig2.18.

Compared with the discrete space collision avoidance of 4 UAVs trajectory shown in Fig.2.19, the trajectory from continuous 3M-RL is smoother.

We also compared continuous 3M-RL with Optimal Reciprocal Collision Avoidance (ORCA) navigation framework, which has been a successful approach to avoid collisions with other moving agents van den Berg *et al.* (2011). ORCA is a heuristic algorithm that agents change their directions when the intruders appear in their "cycle of sensing". However, their methods do not consider the intruders outside the sensing space.

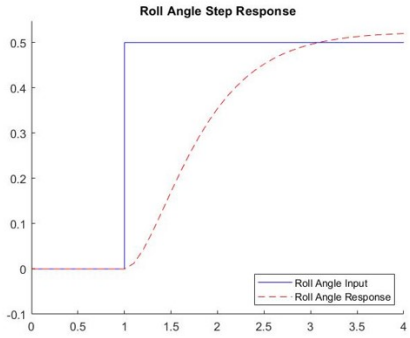



**Figure 2.17:** Roll Angle Step Response in UAVToolBox

The simulation environment of ORCA is also continuous, but the turning action could be any direction, and the heading angle will change as soon as decision being made. Do not consider the intruders outside the sensing space and could not restrain sharp turning makes ORCA suffer from sharp turn, as shown in Fig.2.20. Continuous 3M-RL considers the probability of an intruders get into the viewspace during the training, and is able to get a smoother trajectory.

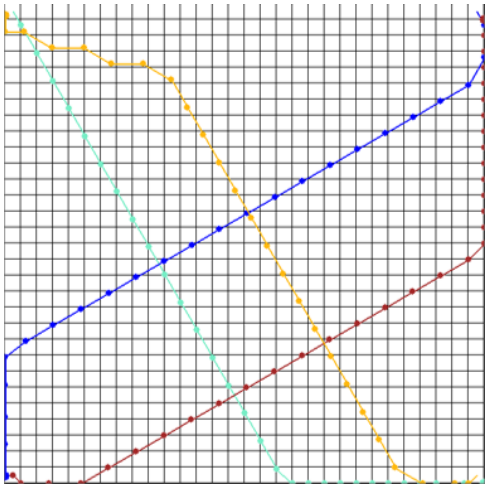


**Figure 2.18:** Continuous 3M-RL Simulation

### *2.3.5 3D Scenario Simulations*

We will present our preliminary simulation results of executing 3M-RL in 3D space. As we mentioned, this simulation is built up on a crude UAV model, the real

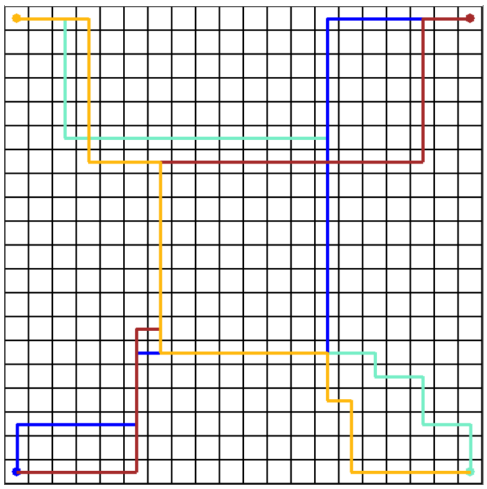


**Figure 2.19:** Discrete 3M-RL Simulation

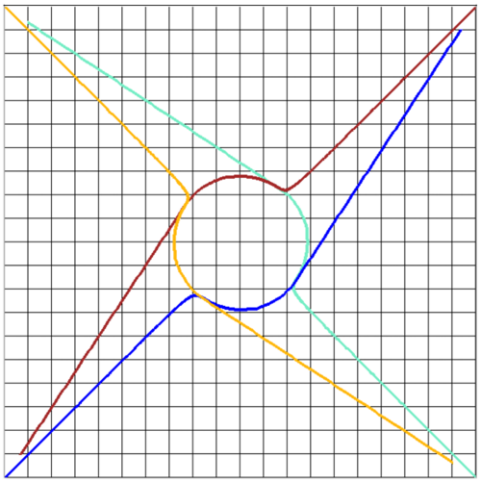


**Figure 2.20:** ORCA Simulation

model largely depends on the control system and dynamics of UAV, we set up this model to test the ability of our algorithm handling collision avoidance problem in 3-D space.

We set up our simulation of 4 UAVs in a $30 \times 30 \times 10$ space, and the collision hull is a cylinder, as shown in Fig.2.9, with radius 5 and height 3. And the detailed view space of observation is the $10 \times 10 \times 5$ space around it. The reward parameters we used here is $c_0 = 0.1, c_1 = 100, c_2 = 1$.

Fig.2.21 shows the result without action reward, although the collision avoidance

problem is considered in 3-D space, 4 UAVs start from one corner and targeting at the point at opposite corner with the same height. As we can see, although the trajectories are free of collisions, the paths are still in a horizontal plane.

Fig.2.22 shows the result with action reward, the rewards of different actions are shown as in 2.3, to encourage the UAVs avoid collisions with different height. The thick blue lines are the planned path of the UAVs, and the red lines on the green plane at the bottom of the figure is the projection of trajectories on the horizontal plane, and this plan is also free of collision.

| Actions | LL | L | M | R | RR | U | D |
|---|---|---|---|---|---|---|---|
| Reward | $-4$ | $-2$ | 0 | $-2$ | $-4$ | $-0.3$ | $-0.3$ |

**Table 2.3:** Details of Actions in 3D Space

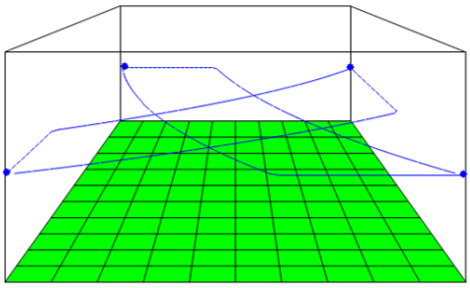


**Figure 2.21:** Simulation of 3M-RL in 3D Space, without Action Reward

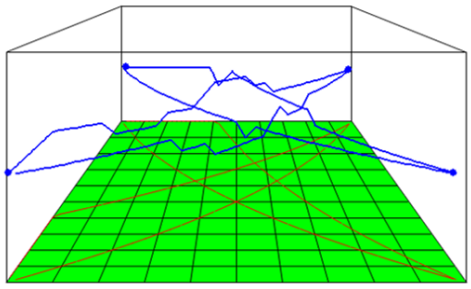


**Figure 2.22:** Simulation of 3M-RL in 3D space, with Action Reward

## 2.4 Extended Multi Agent Collision Avoidance Policy

In previous sections, we have developed a reinforcement learning collision avoidance algorithm, that all agents making decisions on its local state and local observation, which is able to provide a good trajectory, when the start point and destination are fixed. The objective function is defined as

$$\max_{\theta} \mathbb{E}_{a_i(t)\sim\pi^{\theta}(a|s_i(t),o_i(t))}\left[\sum_{t=0}^{\infty} r_i(s_i(t),o_i(t))\middle| s_i(0)=s_i\right] \qquad \text{for all } i \tag{2.10}$$

However, we have found some issue limiting the stability and performance of the algorithm. In this section, we are going to discuss these issues.

### *2.4.1 Reward Function and Nash Equilibrium*

In our previous collision avoidance algorithm, each agent making a decision based on its own position, destination and local observation. The environment returns rewards for different agents. However this setting may not stable or even do not have a Nash equilibrium point. We will used a simple one step toy example to illustrate it. As shown in Tab.2.4 and Tab.2.5, we simply assume it characterizes a one step

| | | $\mathcal{A}_2$ | | |
|---|---|---|---|---|
| | | 1 | 2 | 3 |
| $\mathcal{A}_1$ | 1 | 1 | 2 | -1 |
| | 2 | 2 | 1 | -1 |
| | 3 | -1 | -1 | 3 |

**Table 2.4:** Values of $r_1$ under the Condition of Agent 2's Action

game, and next step of this game is always terminal state no matter what actions they are going to take. Tab.2.4 and Tab.2.5 shows the reward of $r_1$ and $r_2$ when the action of agent 1 and agent 2 are jointly considered. $\mathcal{A}_1 = \{1, 2, 3\}$ is the action space

| | | $\mathcal{A}_2$ | | |
|---|---|---|---|---|
| | | 1 | 2 | 3 |
| $\mathcal{A}_1$ | 1 | 1 | 0 | -1 |
| | 2 | 0 | 1 | -1 |
| | 3 | -1 | -1 | 3 |

**Table 2.5:** Values of $r_2$ under the Condition of Agent 1's Action

of agent 1 and $\mathcal{A}_2$ is the action space of agent 2. It is a one step game, and the reward is also terminal reward. Based on the table, it is clear that there exists an optimal policy for both agent 1 and agent 2, that is $a_1 = 3$ and $a_2 = 3$, and both of the agents would get the largest reward $r_1 = 3$ and $r_2 = 3$. But in multi-agent system that two agents make decisions independently, it is hard to say whether these two agents could converge the optimal actions, or even worse, the policy of these agents will always oscillate in action 1 and 2. The reason is that when $a_1 = 1$, the optimal action of $a_2$ is 1; but when $a_2 = 1$, the optimal action of $a_1$ is 2.

Young (1993) systematically discussed the existence of an equilibrium in an multi-agent game like this. They basically discussed a one step multi-agent game where every agent has its own reward function depends the joint action. They defined a property of this one step game called *weakly acyclic*:

**Definition 1 (Weakly Acyclic Game Young (1993))** *Define the* ***best-reply*** *graph of a game* $\Gamma$ *as follows: each vertex is an* $n$*-tuple of strategy* $\mathbf{a} \in \prod_{i=1}^{n} \mathcal{A}_i$*, where* $\mathcal{A}_i$ *is the strategy set of agent* $i$*. Let* $a_i$ *be the strategy of agent* $i$ *in joint action* $\mathbf{a}$ *and* $a_{-i}$ *be the joint strategy of all other agents. For every two vertex* $\mathbf{a}$ *and* $\mathbf{a}'$*, there is a directed edge* $\mathbf{a} \to \mathbf{a}'$ *if and only if (1)* $\mathbf{a} \neq \mathbf{a}'$ *and (2)there exists exactly one agent* $i$ *such that* $a_i'$ *is the best response to* $a_{-i}$ *on agent* $i$*'s reward function, and* $a_{-i} = a'_{-i}$*. We say the game* $\Gamma$ *is weakly acyclic if its best response graph, from any initial vertex*

*a, there exists a directed path to some vertex $a^*$ from which there is no outgoing edge.*

It is clear that any point without outage edge is an Nash Equilibrium. The best reply graph of the example can be illustrate as in Fig.2.23. As we can see, $(a_1 = 1, a_2 = 1)$, $(a_1 = 1, a_2 = 2)$, $(a_1 = 2, a_2 = 2)$, $(a_1 = 2, a_2 = 1)$ forms a circle, thus is not a Weakly Acyclic Game, which prevent the learning process approaching a Nash equilibrium. We have also verified the oscillation in our previous reward function by using dynamic

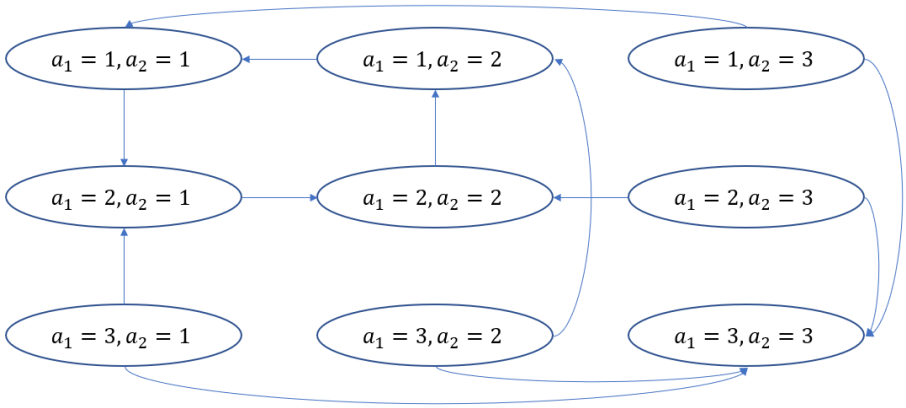



**Figure 2.23:** Best Reply of the Example

programming algorithm. We simulated the 2 agents' collision avoidance problem, even both of agents global state of itself and its intruder, there exists a circle structure in the best-reply graph, in other words, there may not exists a deterministic Nash Equilibrium policy.

In general it is hard to analytically verify whether a game like this has a circle to not, it could be a better solution to use identical reward function to evaluate the joint action of all agents. Zhang *et al.* (2019a) has discussed the format of general cooperative game setting and competitive setting. In general the reward of pure cooperative game has to have identical reward in one time slot.

In our collision avoidance problem, it would be better for as to defined the objective

function as below:

$$\max_{\theta} \mathbb{E}_{a_i(t)\sim\pi^{\theta}(a|s_i(t),o_i(t))}\left[\sum_i \sum_{t=0}^{\infty} r_i(s_i(t),o_i(t))\middle| s_i(0)=s_i\right] \tag{2.11}$$

We still need to mention that multi-agent problems in mean field games like Subramanian and Mahajan (2019), the reward function is defined on different agents, however, it is defined when the distribution of agents get into a stationary distribution as the number of agents goes to infinity. When these assumptions holds, Each agent's effect on the overall multi-agent system can thus become infinitesimal, resulting in all agents being interchangeable and indistinguishable. The uniform value function $V_{\pi,z}(x)$ is then good enough to evaluate the overall performance of the system. As a result, the best reply graph theory does not fit for the mean field games. In this report, we are going to design a policy under the following assumption:

**Assumption 1 (Homogeneous Agent Assumption)** *Every agent takes exactly the same policy $\pi(a_i|s_i)$, where $s_i$ is the state of agent $i$ and $a_i$ is its action.*

This assumption is kind of strong in some scenarios, for example, in our collision avoidance problem if only deterministic policies and deterministic models are considered, agents at the same position and same destination always takes the same action, and can never diverge. We will make some adjustment on the model to deal with it. Consider a swarm control problem under Homogeneous Agent Assumption, methods that takes joint global state for every agent $\mathbf{s} = (s_1, s_2, \cdots, s_n)$ and returns the joint action $\mathbf{a} = (a_1, a_2, \cdots, a_n)$ are redundant. For a homogeneous agents' policy, we do not care about the distribution of the states of the agents, instead of the index of the agents. As a result, our target is to remove the sequence of the joint states and actions.

### 2.4.2 Enhanced Multi-Agent Reinforcement Learning

In this section, we will define our problem in another way, and make it closer to a mean field game that focus the interaction between policy and distribution.

We consider the collision avoidance problem with $N$ agents, $\mathcal{N} := \{1, 2, \cdots, N\}$ is the set of agents. They may have same or different destinations. We first define a state of the $i$th agent in the system $s_i : (pos_i, dest_i) \in \mathcal{S}$ by its position and destination, e.g. $((0,0),(10,10))$ means the agent at position $(0,0)$ and its destination is $(10,10)$, also for simplicity, in general, if the set of destination is limited, we can also use represent destination by its index. In other word words, $((0,0),1)$ means the agent is at position $(0,0)$, heading towards destination 1. Also $\mathcal{S}$ is the state space of one agent. The joint state is $\mathbf{s} = (s_1, s_2, \cdots, s_N) \in \mathcal{S}^N$.

We first consider the collision avoidance problem in discrete grid space, and the action space is defined as

$$\mathcal{A} = \{1 : Up, 2 : Left, 3 : Down, 4 : Right\}.$$

For simplicity, we also assume that the destination of an agent does not change in an episode, also, once the agent reaching its destination, its position could not change any more. Let $\mathbf{a} := (a_1, a_2, \cdots, a_N) \in \mathcal{A}^N$ is the joint action of all agents. The state transition of an agent only depends on the its current state and action of the agent, that is:

$$\begin{aligned} P(\mathbf{s}(t+1)|\mathbf{s}(t), \mathbf{a}(t)) &= \prod_{i=1}^{N} P(s_i(t+1)|\mathbf{s}(t), \mathbf{a}(t)) \\ &= \prod_{i=1}^{N} P(s_i(t+1)|s_i(t), a_i(t)) \end{aligned}$$

To solve the problem of collision avoidance, we defined the reward function on joint state and action as below:

$$
\begin{aligned}
r_i(\mathbf{s}, \mathbf{a}) &= -\alpha \cdot dist_d(s_i) - \beta \mathbb{1}\{\exists j, dist_p(s_i, s_j) \leq dist_{safe}\} \\
r(\mathbf{s}, \mathbf{a}) &= \sum_{i=1}^{N} r_i(\mathbf{s}, \mathbf{a})
\end{aligned} \tag{2.12}
$$

where $dist_d$ is the destination distance function, returns the distance from the position to the destination in $s_i$, $dist_p$ is the position distance function, it returns the distance between the positions, $dist_{safe}$ is the safe distance required by the application and $\beta$ is the penalty of collision. $r_i$ can be understood as the reward on agent $i$, however, all agents need to optimized the overall reward $r$, instead of its own reward. And it can be easily understood that $r_i$ is bounded in finite space, we assume $|r_i(\mathbf{s}, \mathbf{a})| < M$. Also here we assume that $r_i$ only depends on intruders around agent $i$ and its own action $a_i$, detailed global state and actions of other agents are not required.

With the Assumption.1, the interaction the index of different agents are not importance, to characterize it, we define a $\{\mathbf{s}\}$ as the set of joint state, removing the index and sequence of the elements in it. Then all agents are associated with a uniform localized policy $\pi^\theta$ parameterized by $\theta$. The localized policy $\pi^\theta(a_i|s_i, O(s_i, \{\mathbf{s}\}))$ is the distribution on the action $a_i$ for the agent $i$ whose state is $s_i$, conditioned on an **anonymous observation** $O(s_i, \{\mathbf{s}\})$ on its neighborhood, where $O$ is the obervation function on set $\{s\}$ around state $s_i$. We will define the observation function in the following section. We also need to notice that if $s \notin \{\mathbf{s}\}$, the policy function $\pi$ is meaningless. Then we use

$$
\boldsymbol{\pi}^\theta(\mathbf{a}|\mathbf{s}) = \prod_{i=1}^{N} \pi^\theta(a_i|s_i, O(s_i, \{s\}))
$$

to denote the joint policy, which is the product distribution of the localized policies as each agent acts independently. By doing this, the complex interaction between every

agent to all the other agents can be characterized by the interaction between agent and the corresponding distribution of global state over state space $\mathcal{S}$ like the mean field game in Subramanian and Mahajan (2019). The main reason we using anonymous observation is that the complexity can be largely reduces under Assumption.1, we will discuss the details in the following sections.

The objective is to find the best homogeneous policy $\pi^\theta$ such that the discounted global stage reward is maximized, starting from some initial state distribution $\mathbf{s}_0$,

$$\max_\theta J(\theta) := \mathbb{E}_{s\sim s_0}\mathbb{E}_{\mathbf{a}(t)\sim\boldsymbol{\pi}^\theta(\cdot|\mathbf{s}(t))}\left[\sum_{t=0}^{\infty}\gamma^t r(\mathbf{s}(t),\mathbf{a}(t))\middle|\mathbf{s}(0)=s\right] \tag{2.13}$$

Also, for a fixing localized homogeneous policy $\pi^\theta$, the Q-function for this policy is defined as:

$$\begin{aligned} Q^\theta(\mathbf{s},\mathbf{a}) &:= \mathbb{E}_{\mathbf{a}(t)\sim\boldsymbol{\pi}^\theta(\cdot|\mathbf{s}(t))}\left[\sum_{t=0}^{\infty}\gamma^t r(\mathbf{s}(t),\mathbf{a}(t))\middle|\mathbf{s}(0)=\mathbf{a},\mathbf{a}(0)=\mathbf{a}\right] \\ &= \sum_{i=1}^{N}\mathbb{E}_{\mathbf{a}(t)\sim\boldsymbol{\pi}^\theta(\cdot|\mathbf{s}(t))}\left[\sum_{t=0}^{\infty}\gamma^t r_i(\mathbf{s}(t),\mathbf{a}(t))\middle|\mathbf{s}(0)=\mathbf{a},\mathbf{a}(0)=\mathbf{a}\right] \\ &= \sum_{i=1}^{N} Q_i^\theta(\mathbf{s},\mathbf{a}) \end{aligned} \tag{2.14}$$

### *2.4.3 Exponential Decay of Q-function Leads to Approximation on Local Observation*

The idea of this part comes from Qu *et al.* (2020), they discussed the approximation in network applications, and the set of neighborhood is fixed, here we are going to extend it into dynamic neighbors under the Assumption.1. In this section, we assume that the observation $O(s_i,\{\mathbf{s}\})$ includes states of all agents in the observable region, as shown in the red blocks in Fig.2.24. The observable region is the defined as the area where the intruders in it can reach the position of $s_i$ within $d_{obs}$ steps. As shown in Fig.2.24, $d_{obs}=2$. We also assume that risk region is contained in the observable region.

Define $\mathcal{N}_s^k(\mathbf{s})$ to be the set of agents that could get into the observable region of agents in state $s$ within $k$-step when the global state is $\mathbf{s}$, and $\mathcal{N}_s^0(\mathbf{s})$ is the set of agents in the observable region of state $s$ including it self. For example, Fig.2.24 illustrates a simple example of set of neighbors when safe distance is 2. We simply assume the black agent is in state $s$. The only neighbor in its 0 step neighborhood is itself. The blue agent is in its 4 step neighborhood. We also defined $\mathcal{N}_{-s}^k(\mathbf{s}) = \mathcal{N} \setminus \mathcal{N}_s^k(\mathbf{s})$ as the set of agents out side its $k$ step neighborhood. The intuition of neighborhood is that intruders outside an agent's $k$ step neighborhood will not get into its risk region until $k/2$ step in the future, and the collision penalty induced by this intruder will always has a discounted factor at least $\gamma^{k/2+1}$. Next, we will prove this property in equations.

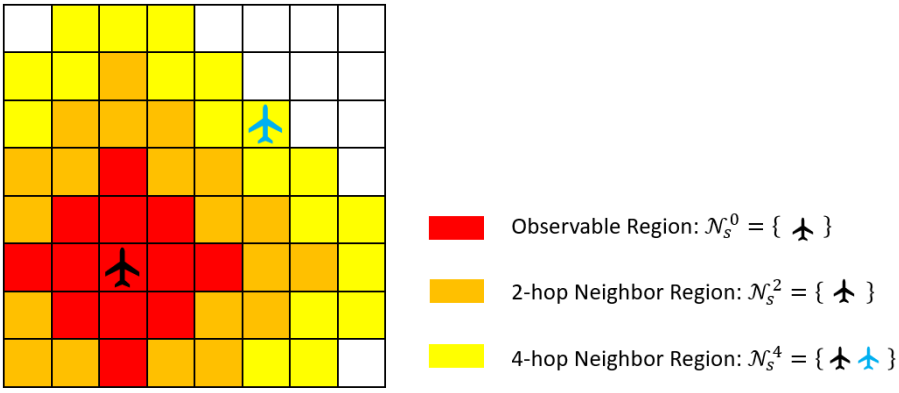



**Figure 2.24:** Set of Neighbors

Let the state of agent $i$ to be $s_i$, and the global state is $\mathbf{s}$. The global state can also be written as $(\mathbf{s}_{\mathcal{N}_{s_i}^k}, \mathbf{s}_{\mathcal{N}_{-s_i}^k})$, where $\mathbf{s}_{\mathcal{N}_{s_i}^k}$ is the joint state of agents in agent $i$'s k step neighborhood, to avoid the abuse of notation, we omit the brackets in neighbor function $\mathcal{N}_s^k$. Similarly, we write $(\mathbf{a}_{\mathcal{N}_{s_i}^k}, \mathbf{a}_{\mathcal{N}_{-s_i}^k})$ as the joint action. Then we could have following property, the intuition of the theorem come from Qu *et al.* (2020).

**Theorem 1 (Extended Exponential Decay Property)** *The $(c, \rho)$-exponential decay property holds if, for any localized homogeneous policy $\pi^\theta$, for any $s \in \mathcal{N}$,*

$\mathbf{s}_{\mathcal{N}_{s_i}^k} \in \mathcal{S}^{|\mathcal{N}_{s_i}^k|}$ , $\mathbf{s}_{\mathcal{N}_{-s_i}^k}, \mathbf{s}'_{\mathcal{N}_{-s_i}^k} \in \mathcal{S}^{|\mathcal{N}_{-s_i}^k|}$, $\mathbf{a}_{\mathcal{N}_{s_i}^k} \in \mathcal{A}^{|\mathcal{N}_{s_i}^k|}$ , $\mathbf{a}_{\mathcal{N}_{-s_i}^k}, \mathbf{a}'_{\mathcal{N}_{-s_i}^k} \in \mathcal{A}^{|\mathcal{N}_{-s_i}^k|}$, $Q_i^\theta$ *satisfies,*

$$|Q_i^\theta(\mathbf{s}_{\mathcal{N}_{s_i}^k}, \mathbf{s}_{\mathcal{N}_{-s_i}^k}, \mathbf{a}_{\mathcal{N}_{s_i}^k}, \mathbf{a}_{\mathcal{N}_{-s_i}^k}) - Q_i^\theta(\mathbf{s}_{\mathcal{N}_{s_i}^k}, \mathbf{s}'_{\mathcal{N}_{-s_i}^k}, \mathbf{a}_{\mathcal{N}_{s_i}^k}, \mathbf{a}'_{\mathcal{N}_{-s_i}^k})| \leq c\rho^{\lfloor k/(d_{obs}+2)\rfloor} \tag{2.15}$$

**Proof:** From the left hand side of Eq.2.15, we have:

$$\begin{aligned}
&|Q_i^\theta(\mathbf{s}_{\mathcal{N}_{s_i}^k}, \mathbf{s}_{\mathcal{N}_{-s_i}^k}, \mathbf{a}_{\mathcal{N}_{s_i}^k}, \mathbf{a}_{\mathcal{N}_{-s_i}^k}) - Q_i^\theta(\mathbf{s}_{\mathcal{N}_{s_i}^k}, \mathbf{s}'_{\mathcal{N}_{-s_i}^k}, \mathbf{a}_{\mathcal{N}_{s_i}^k}, \mathbf{a}'_{\mathcal{N}_{-s_i}^k})| \\
\leq &\sum_{t=0}^{\infty} |\mathbb{E}_{(\mathbf{s}(t),\mathbf{a}(t))\sim\pi^\theta}[\gamma^t r_i(\mathbf{s},\mathbf{a})|(\mathbf{s}(0),\mathbf{a}(0)) = (\mathbf{s},\mathbf{a})] \\
&- \mathbb{E}_{(\mathbf{s}(t),\mathbf{a}(t))\sim\pi^\theta}[\gamma^t r_i(\mathbf{s},\mathbf{a})|(\mathbf{s}(0),\mathbf{a}(0)) = (\mathbf{s}',\mathbf{a}')]|
\end{aligned}$$

The equation holds since $|\sum_i x_i| \leq \sum_i |x_i|$.

Next, we start from $k = 0$. When $k = 0$, $s_i$ and states of agents in $\mathcal{N}_{s_i}(\mathbf{s})$ are equal in $\mathbf{s}$ and $\mathbf{s}'$. As shown in Fig.2.25, the observable space of agent $i$ in $\mathbf{s}$ and $\mathbf{s}'$ are exactly the same. As a result, the policy and reward of agent $i$ are also equal in $\mathbf{s}$ and $\mathbf{s}'$, and we can have

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{s}(t),\mathbf{a}(t))\sim\pi^\theta}[r_i(\mathbf{s}(0),\mathbf{a}(0))|(\mathbf{s}(0),\mathbf{a}(0)) = (\mathbf{s},\mathbf{a})] = \\
&\qquad \mathbb{E}_{(\mathbf{s}(t),\mathbf{a}(t))\sim\pi^\theta}[r_i(\mathbf{s}(0),\mathbf{a}(0))|(\mathbf{s}(0),\mathbf{a}(0)) = (\mathbf{s}',\mathbf{a}')]
\end{aligned} \tag{2.16}$$

On the other hand, as we can see in Fig.2.25, there will be an intruder get into agent

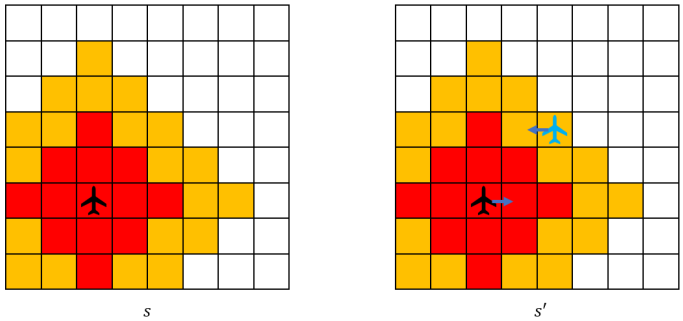



**Figure 2.25:** k=0

$i$'s observable space in state $\mathbf{s}'$, but not in $\mathbf{s}$. We also have

$$|\mathbb{E}_{(\mathbf{s}(t),\mathbf{a}(t))\sim\pi^\theta}[\gamma^t r_i(\mathbf{s}(t),\mathbf{a}(t))|(\mathbf{s}(0),\mathbf{a}(0)) = (\mathbf{s},\mathbf{a})] - \mathbb{E}_{(\mathbf{s}(t),\mathbf{a}(t))\sim\pi^\theta}[\gamma^t r_i(\mathbf{s}(t),\mathbf{a}(t))|(\mathbf{s}(0),\mathbf{a}(0)) = (\mathbf{s}',\mathbf{a}')]| < 2M\cdot\gamma^t$$

for $t \geq 1$. As a result, we have:

$$|Q_i^\theta(\mathbf{s}_{\mathcal{N}_{s_i}^k}, \mathbf{s}_{\mathcal{N}_{-s_i}^k}, \mathbf{a}_{\mathcal{N}_{s_i}^k}, \mathbf{a}_{\mathcal{N}_{-s_i}^k}) - Q_i^\theta(\mathbf{s}_{\mathcal{N}_{s_i}^k}, \mathbf{s}'_{\mathcal{N}_{-s_i}^k}, \mathbf{a}_{\mathcal{N}_{s_i}^k}, \mathbf{a}'_{\mathcal{N}_{-s_i}^k})| \leq \gamma\frac{2M}{1-\gamma}$$

for $k \geq 0$. Next we consider the largest $k$ that is possible to have

$$\mathbb{E}_{(\mathbf{s}(t),\mathbf{a}(t))\sim\pi^\theta}[\gamma^t r_i(\mathbf{s}(1),\mathbf{a}(1))|(\mathbf{s}(1),\mathbf{a}(1)) = (\mathbf{s},\mathbf{a})] \neq \mathbb{E}_{(\mathbf{s}(t),\mathbf{a}(t))\sim\pi^\theta}[\gamma^t r_i(\mathbf{s}(1),\mathbf{a}(1))|(\mathbf{s}(1),\mathbf{a}(1)) = (\mathbf{s}',\mathbf{a}')] \tag{2.17}$$

under some $\pi^\theta$. As shown in Fig.2.26, it illustrates the comparison of two scenarios $\mathbf{s}$ and $\mathbf{s}'$ at $t = 0$ and $t = 1$, where the black agent is regards as agent $i$. We assume in both of these scenarios, $k = 1 + d_{obs}$, and the rewards of these two scenarios are $r$ and $r'$ respectively. First consider $\mathbf{s}$ case, we assume that under some policy $\pi^\theta$, both blue and black agents are taking action "Right" with probability 1 when $t = 0$. As $t = 1$, the blue agent is still outside the observable area of the black agent. Then we consider $\mathbf{s}'$ case, compare with $\mathbf{s}$, because of the existence of green agent which is in the observable area of the blue agent, we assume that under policy $\pi^\theta$, the action of blue agent is modified to taking "Left" with probability 1. As a result, at $t = 1$, the blue agent is in the observable area of black agent, and $r'(1) \neq r(1)$.

On the other hand, if $k = 2 + d_{obs}$, the observable area of all agents that is possible to get into the observable are of the black agent is contained in $\mathcal{N}_{s_i}^{2+d_{obs}}$, and it can be guaranteed to have

$$\mathbb{E}_{(\mathbf{s}(t),\mathbf{a}(t))\sim\pi^\theta}[\gamma^t r_i(\mathbf{s}(1),\mathbf{a}(1))|(\mathbf{s}(0),\mathbf{a}(0)) = (\mathbf{s},\mathbf{a})] = \mathbb{E}_{(\mathbf{s}(t),\mathbf{a}(t))\sim\pi^\theta}[\gamma^t r_i(\mathbf{s}(1),\mathbf{a}(1))|(\mathbf{s}(0),\mathbf{a}(0)) = (\mathbf{s}',\mathbf{a}')]. \tag{2.18}$$

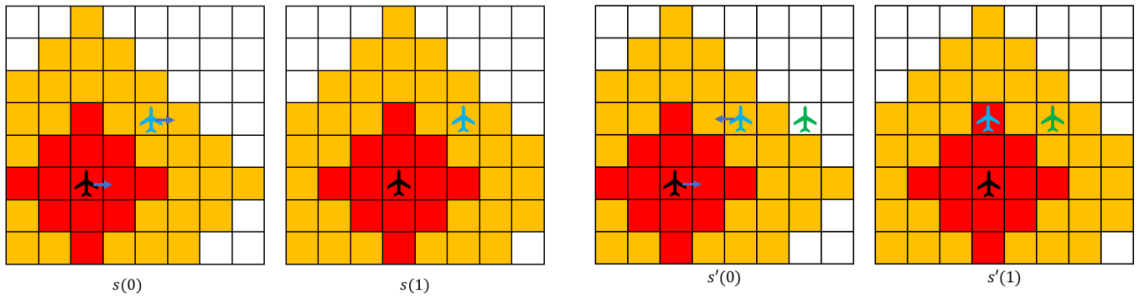



**Figure 2.26:** Largest $k$ Satisfying Eq.2.16 and Eq.2.17.

As a result, we could have

$$|Q_i^\theta(\mathbf{s}_{\mathcal{N}_{s_i}^k}, \mathbf{s}_{\mathcal{N}_{-s_i}^k}, \mathbf{a}_{\mathcal{N}_{s_i}^k}, \mathbf{a}_{\mathcal{N}_{-s_i}^k}) - Q_i^\theta(\mathbf{s}_{\mathcal{N}_{s_i}^k}, \mathbf{s}'_{\mathcal{N}_{-s_i}^k}, \mathbf{a}_{\mathcal{N}_{s_i}^k}, \mathbf{a}'_{\mathcal{N}_{-s_i}^k})| \leq \gamma^2 \frac{2M}{1-\gamma}$$

for $k \geq 2 + d_{obs}$.

Similarly, we next consider the largest $k$ that is possible to satisfy Eq.2.16, 2.18 and following Eq.2.19.

$$\mathbb{E}_{(\mathbf{s}(t),\mathbf{a}(t))\sim\pi^\theta}[\gamma^t r_i(\mathbf{s}(2),\mathbf{a}(2))|(\mathbf{s}(0),\mathbf{a}(0)) = (\mathbf{s},\mathbf{a})] \neq \tag{2.19}$$
$$\mathbb{E}_{(\mathbf{s}(t),\mathbf{a}(t))\sim\pi^\theta}[\gamma^t r_i(\mathbf{s}(2),\mathbf{a}(2))|(\mathbf{s}(0),\mathbf{a}(0)) = (\mathbf{s}',\mathbf{a}')]$$

We are going to illustrate it with another example as shown in Fig.2.27, and for simplicity, we just show it in 2-D space. In these two scenarios, $k = 3 + 2d_{obs}$. Compare with $\mathbf{s}$ case, because of the existence of green intruder in $\mathbf{s}'$ case, the action of intruder 2 is changing from "Right" $w.p.1$ to "Left" $w.p.1$ as $t = 0$. As a result, at $t = 1$, intruder 2 get into the observable area of intruder 1, which change the action of intruder 1 from "Right" $w.p.1$ to "Left" $w.p.1$ as $t = 1$. This is exactly same as $t = 0$ in Fig.2.26, and for the same reason, intruder 1 get into the observable area of agent $i$ when $t = 2$, which makes $r'(2) \neq r(2)$.

On the other hand, if $k \geq 4 + 2d_{obs}$, it can be guaranteed that the observable area of both intruder 1 and 2 are contained in $\mathcal{N}_{s_i}^{4+2d_{obs}}$, in other words, the observable area of all intruders that could get into the observable area of agent $i$ before $t = 2$ is

contained in $\mathcal{N}_{s_i}^{4+2d_{obs}}$. As a result, we could have:

$$|Q_i^\theta(\mathbf{s}_{\mathcal{N}_{s_i}^k}, \mathbf{s}_{\mathcal{N}_{-s_i}^k}, \mathbf{a}_{\mathcal{N}_{s_i}^k}, \mathbf{a}_{\mathcal{N}_{-s_i}^k}) - Q_i^\theta(\mathbf{s}_{\mathcal{N}_{s_i}^k}, \mathbf{s}'_{\mathcal{N}_{-s_i}^k}, \mathbf{a}_{\mathcal{N}_{s_i}^k}, \mathbf{a}'_{\mathcal{N}_{-s_i}^k})| \leq \gamma^3 \frac{2M}{1-\gamma}$$

for $k \geq 4 + 2d_{obs}$. At last, with induction, we could get the following equation

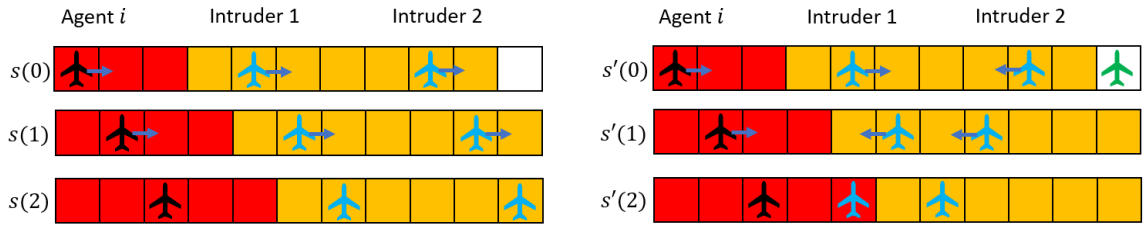



**Figure 2.27:** Largest $k$ Satisfying Eq.2.16,2.18,2.19.

$$|Q_i^\theta(\mathbf{s}_{\mathcal{N}_{s_i}^k}, \mathbf{s}_{\mathcal{N}_{-s_i}^k}, \mathbf{a}_{\mathcal{N}_{s_i}^k}, \mathbf{a}_{\mathcal{N}_{-s_i}^k}) - Q_i^\theta(\mathbf{s}_{\mathcal{N}_{s_i}^k}, \mathbf{s}'_{\mathcal{N}_{-s_i}^k}, \mathbf{a}_{\mathcal{N}_{s_i}^k}, \mathbf{a}'_{\mathcal{N}_{-s_i}^k})| \leq \gamma^{m+1} \frac{2M}{1-\gamma}$$

for $k \geq m(2 + d_{obs})$, and theorem proved. □

### *2.4.4 New Algorithm*

Based on the Exponential Decay Property we have proved, we can defined our algorithm as shown in Alg.2. The intuition of this algorithm is that when $k$ is large enough, the influence of intruders outside $\mathcal{N}_{s_i}^k$ to $Q_i$ can be ignored. On the other hand, the policy of agent $i$ at state $s_i$ is almost irrelevant with $Q_i$ for agent $j$ outside $\mathcal{N}_{s_i}^k$. That is saying

$$\begin{aligned}\frac{\partial Q(\mathbf{s}, \mathbf{a})}{\partial \pi(a_i|s_i, o_i)} &= \frac{\partial \sum_j Q_j(\mathbf{s}, \mathbf{a})}{\partial \pi(a_i|s_i, o_i)} \\ &\approx \frac{\partial \sum_{j \in \mathcal{N}_{s_i}^k} Q_j(\mathbf{s}, \mathbf{a})}{\partial \pi(a_i|s_i, o_i)}\end{aligned}$$

### *2.4.5 Analysis and Contribution*

The In this section, we derived a extended exponential decay property based on local observable algorithm and dynamic neighborhood. The property states that

**Algorithm 2:** Algorithm

**Input:** A policy parameterization $\pi^{\boldsymbol{\theta}}(a|s_i, o_i)$ and a state value parameterization $\hat{V}^{\mathbf{w}}(\mathbf{s}_{\mathcal{N}_{s_i}^k})$

**foreach** *Episode* **do**

  **while** *Any Agent Not Arrived and* $t < T$ **do**

    Take action $a(t) \sim \pi^{\boldsymbol{\theta}}(\cdot|s_i(t), o_i(t))$ for all $i$;

    Get next relative position $s_i(t+1)$, next observation $o_i(t+1)$, and reward $r_i(t)$ for all $i$;

    Record $\big(s_i(t), o_i(t), \mathbf{s}_{\mathcal{N}_{s_i}^k}(t), a_i(t), r_i(t), s_i(t+1), o_i(t+1)\big)$ for all $i$

  **end**

  **foreach** $t$ *in the episode* **do**

    $\delta_i \leftarrow r_i(t) + \gamma \hat{V}^{\mathbf{w}}(\mathbf{s}_{\mathcal{N}_{s_i}^k}(t+1)) - \hat{V}^{\mathbf{w}}(\mathbf{s}_{\mathcal{N}_{s_i}^k}(t))$ for all $i$;

    Network parameters update: $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \frac{1}{n} \sum_i \delta_i \nabla \hat{V}^{\mathbf{w}}(\mathbf{s}_{\mathcal{N}_{s_i}^k})$;

    Network parameters update:

    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} \frac{1}{n} \sum_i \frac{1}{|\mathcal{N}_{s_i}^k|} \sum_{j \in \mathcal{N}_{s_i}^k} \delta_j \nabla \ln \pi^{\boldsymbol{\theta}}(a(t)|s_i(t), o_i(t))$

  **end**

**end**

intruders' influence on $Q_i$, the reward-to-go of agent $i$, decay exponentially $w.r.t$ the distance between the position of agent $i$ and the intruder. Based on that, we developed a centralized training with decentralized execution algorithm Alg.2 to solve the objective function Eq.2.11. The diagram of our algorithm is shown as in Fig.2.30. Compare with general centralized training with decentralized execution algorithms as shown in Fig.2.29, we have used a decentralized critic network and largely reduce the complexity.

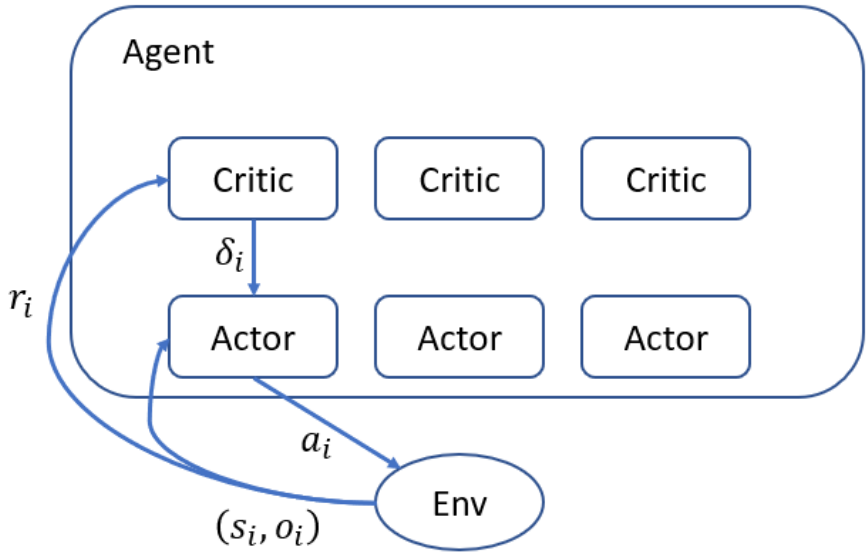



**Figure 2.28:** Previous algorithm, unstable

### *2.4.6 Simulation Results*

In this section, we basically compare the simulation results of our algorithm in Alg.2 and Fig.2.30 and our previous algorithm illustrated in Fig.2.28. The comparison of total rewards in different scenarios are shown in Fig.2.31,2.32,2.33. In these plot, our new algorithm is slower than our previous algorithm, however since these multi-agent reinforcement learning algorithms do not even have stability guarantee, the speed of training is not that important. On the other hand, it looks that the difference of total reward between these two algorithms are not apparent, the reason is that the total reward are dominated by the terminal reward and penalty of collision, compare with a random walk algorithm at the first episode, both algorithm could avoid the collision and reach the destination with a relative good policy.

The values of total reward in these scenarios are shown in Tab.2.4.6. The results show that our new algorithm can get a better performance than the previous algorithm even if the policy space remains the same.

| Grid Size | Num of Agents | Safe Distance | Old Alg | new Alg |
|---|---|---|---|---|
| 30 | 4 | 2 | -2.7519 | -2.2435 |
| 30 | 8 | 2 | -8.3003 | -7.8122 |
| 30 | 8 | 3 | -7.3276 | -3.4722 |

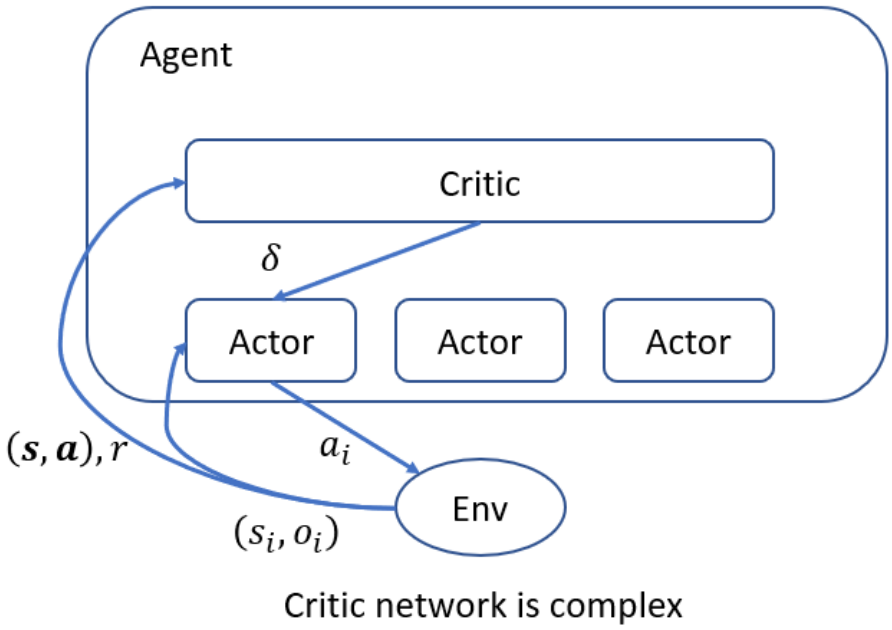



**Figure 2.29:** General Centralized Training with Decentralized Execution Algorithms, Critic Network Has Large Complexity

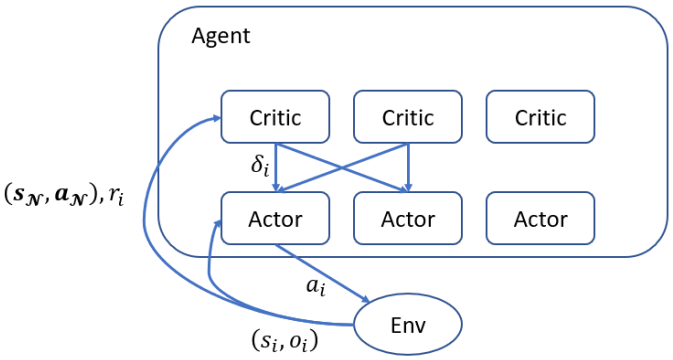



**Figure 2.30:** New Algorithm

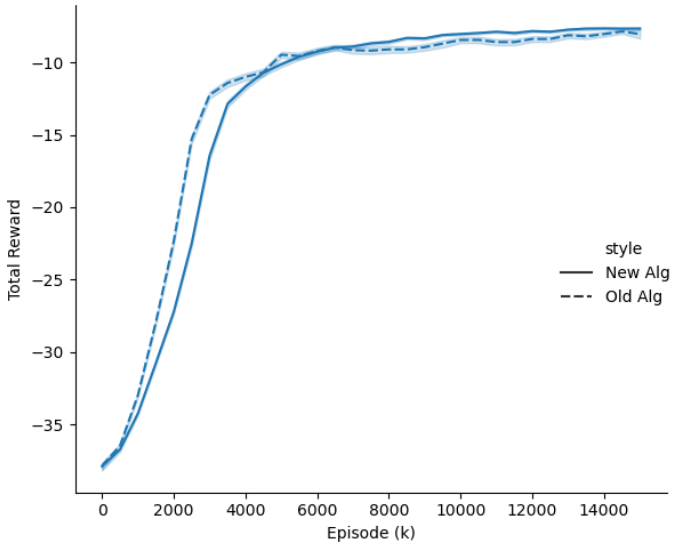



**Figure 2.31:** Comparison of Total Reward of 4 Agents in 30 Grid Size, Safe Distance=2

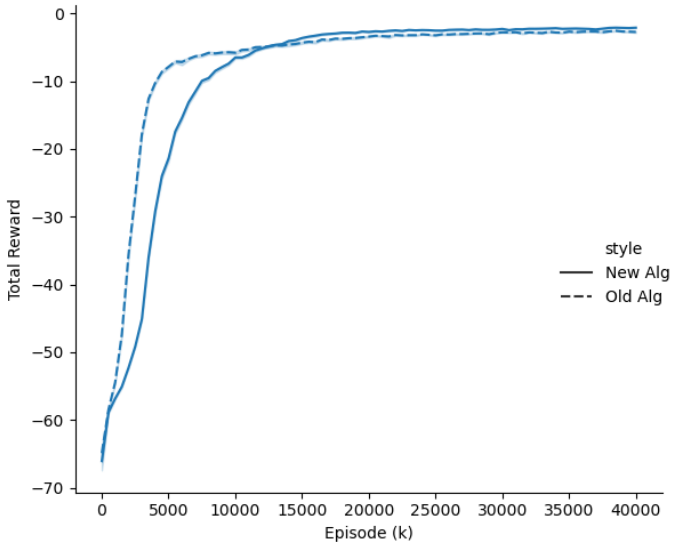



**Figure 2.32:** Comparison of Total Reward of 8 Agents in 30 Grid Size, Safe Distance=2

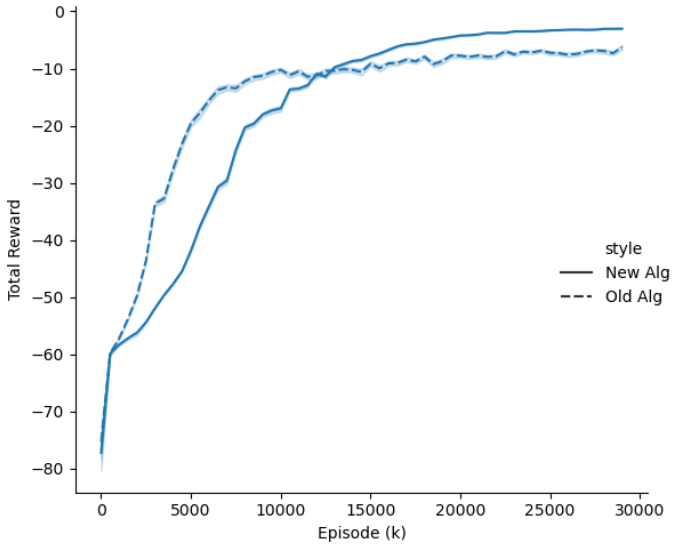



**Figure 2.33:** Comparison of Total Reward of 8 Agents in 30 Grid Size, Safe Distance=3

# Chapter 3

# LEARNING PARALLEL MARKOV CHAINS OVER UNRELIABLE WIRELESS CHANNELS

In this chapter, we are going to discussed the problem that the central controller needs to maintain an estimate of the risk level of each UAV. When a report from an UAV is successfully is received, the state of the Markov chain is know; otherwise, the estimate of the distribution of the risk level of an UAV is updated based on a pre-defined Markov chain. In this chapter, we assume simple two-state Markov chains. Due to limited bandwidth, the controller can only probe a subset of Markov chains each time. The objective is to develop a scheduling algorithm to minimize the total information entropy of the Markov chains.

This optimization problem is then formulated as a Multi-Armed Bandit (MAB) problem with the capacity of wireless channels as a hard constraint. The problem is similar to a restless bandit problem. The key difference is that the objective is to minimize the total information entropy of all bandits instead of finiding the optimal bandit. We adopt Whittle's Index to solve the problem. Whittle's Index was first proposed in Whittle (1988) for restless bandit problems. Whittl's index has been used in wireless communication problems. For example, Anand and de Veciana (2018) consider sa delay minimization problem through a multi-state channels, and Liu and Zhao (2010) studies the throughput maximization problem where transmitter in the system has dynamic multi-channel access.

In this chapter, we consider both single-rate wireless channels and multi-rate wireless channels. We prove that the problem is indexable for single-rate wireless channels and establish a sufficient condition under which the problem is indexable with multi-

rate wireless channels. Our numerical evaluations show that our algorithm outperform other heuristics such as the greedy policy and Round& Robin policies.

## 3.1 Problem Formulation

We consider a system consisting of $M$ two-state Markov chains as shown in Fig.3.1. For simplicity, we assume that for the $i$th Markov chain, $p_{10} = p_{01} = p_i < 0.5$.

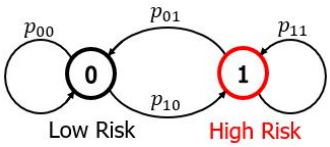



**Figure 3.1:** A Two-State Markov Chain

We assume the controller can probe at most $K$ $(K < M)$ of them at each time slot, and each probe succeeds with probability $r < 1$. Let $S_i(t)$ denote the state of the Markov chain $i$ at time $t$, and $\theta_i(t) \in [0, 1]$ denotes the probability that Markov chain $i$ is in state "1" at time slot $t$ given the most recent observation received at the controller.

$$\theta_i(t) = \begin{cases} S_i(t), & \text{if the probe is successful} \\ p_i + (1 - 2p_i)\theta_i(t-1), & \text{otherwise} \end{cases} \tag{3.1}$$

Given a Bernoulli distribution with parameter $\theta_i(t)$, the entropy of the distribution is

$$c_i(t) = -\theta_i(t) \log \theta_i(t) - (1 - \theta_i(t)) \log(1 - \theta_i(t)).$$

The problem we are interested in is to minimize the overall entropy of the system, i.e.

$$\begin{aligned} &\min_{\pi \in \Pi} E\left[\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{M} c_i(t)\right] \\ &\text{subject to:} \sum_{i=1}^{M} A_i^{\pi}(t) \leq K, \quad \forall t, \end{aligned} \tag{3.2}$$

where

$$A_i^\pi(t) = \mathbb{1}\{\text{Markov chain } i \text{ is probed at time } t\}$$

under a scheduling policy $\pi$, and $\Pi$ is the set of all scheduling policies.

If we view $\theta_i(t)$ as the state of an arm, then the problem is related to restless bandit problems. A significant difference is that the reward $\sum_i c_i(t)$ depends on the states of all arms.

We use Whittle's index Whittle (1988) to solve this problem. We will see that despite the fundamental difference in the cost function, the problem is an indexable problem. Following Whittle's index approach, we first relax the hard constraint per time slot to an average constraint, i.e., the number of Markov chains to be obsrved is at most $K$ on average,

$$\sum_{t=0}^{\infty}\sum_{i=1}^{M}\beta^t A_i^\pi(t) \le \frac{K}{1-\beta}.$$

By introducing the Lagrange multiplier $v$ to the problem, we have the following Lagrangian:

$$\mathcal{L}(v) = \min_{\pi\in\Pi} E\left[\sum_{t=0}^{\infty}\beta^t\sum_{i=1}^{M}c_i(t) + v\sum_{t=0}^{\infty}\beta^t\left(\sum_{i=1}^{M}A_i^\pi(t) - K\right)\right] \tag{3.3}$$

Note that the Lagrange multiplier $v$ can be viewed as the penalty. Since the term $v\sum_{t=0}^{\infty}\beta^t K$ is a constant in the optimization problem, for a fixed $v$, the relaxed problem can be decoupled into subproblems associate with each individual Markov chain. In particular, we have

$$\min_{\pi\in\Pi} E\left[\sum_{t=0}^{\infty}\beta^t c_i(t) + v\sum_{t=0}^{\infty}\beta^t A_i^\pi(t)\right]. \tag{3.4}$$

Note that while we replace the hard constraint, the algorithm implemented can only probe $K$ Markov chains. The Whittle index approach is to index the $M$ Markov chains and then the algorithm picks the $K$ ones with the highest indices.

The Whittle's Index Policy is a low-complexity heuristic that has been extensively used in the literature and performs well in practice.The challenge is that problems are not always indexable. In the following sections, we will prove the indexability and the conditioned indexability, i.e. Whittle's index is well defined.

## 3.2 Whittle's Index Approach

To solve the sub-problem Eq.(3.4) for each Markov chain. We consider the following Bellman equation:

$$\begin{aligned} V_i(\theta_i; v) = \min \Big\{ & c_i(\theta_i) + \beta V_i\big(p_i + (1-2p)\theta_i; v\big), v + \\ & c_i(\theta_i) + \beta \Big[ r\theta_i V_i(1; v) + r(1-\theta_i) V_i(0; v) + \\ & (1-r) V_i\left(p_i + (1-2p_i)\theta_i; v\right) \Big] \Big\} \end{aligned} \tag{3.5}$$

where $V_i(\theta, v)$ is the value function of the $i$th Markov chain starting from state $\theta_i(t)$, and $r$ is the message delivery ratio. The Whittle index in this problem is $v^*(\theta_i, r)$, the smallest value of $v$ in the Eq. (3.4) that makes it equally desirable to observe and not to observe when the $i$th Markov chain is in state $\theta_i$. The fundamental question in Whittle's index whether the problem is indexable. We will analyze the indexability in two different cases.

### *3.2.1 Single State Channels*

We first consider the case the message delivery ratio $r$ remains the same at all time, and have the following lemma.

**Lemma 1** *$V_i(\theta_i; v)$ is a concave function in $\theta_i$.*

The proof of this lemma can be found in the Appendix.

Letting the two terms in the minimization equal to each other, we obtain

$$\begin{aligned} c_i(\theta_i(t)) + \beta V_i\big(p_i + (1-2p_i)\theta_i(t); v\big) &= v + c_i(\theta_i(t)) \\ \beta r\theta_i(t) V_i(1; v) + \beta r(1-\theta_i(t)) V_i(0; v) + & \\ \beta(1-r) V_i\left(p + (1-2p_i)\theta_i(t); v\right). & \end{aligned} \tag{3.6}$$

Since the cost entropy function is symmetric in $\theta$, we know that $V_i(0; v) = V_i(1; v)$, which yields

$$\beta r V_i\big(p_i + (1-2p_i)\theta_i(t); v\big) = v + \beta r V_i(0; v). \tag{3.7}$$

We next show that the problem is indexable when $r$ is given. Let $D_i(v)$ be the set of values of $\theta_i$ for which Markov chain $i$ will not be probed under the $v$-penalty policy, i.e.

$$D_i(v) = \{\theta \in [0,1] : \beta r V_i\big(p_i + (1-2p_i)\theta; v\big) < v + \beta r V_i(0; v)\}.$$

The problem is indexable if $D_i(v)$ increases monotonically from $\varnothing$ to the universe set as $v$ increasing from 0 to $\infty$, as established in the following theorem.

**Theorem 2** *The Markov chains are indexable.*

**Proof:** The indexable condition is equivalent to that Equation (3.7) has a unique solution $v^*(\theta)$ for each state $\theta$. Because of the symmetry of the cost function, we know that $V_i(0; v) = V_i(1; v)$. Based on the concavity of $V_i(\theta; v)$, we have

$$V_i\big(p_i + (1-2p_i)\theta_i; v\big) > V_i(0; v)$$

because for every $\theta \in (0,1)$, the point $(\theta, V_i(\theta; v))$ on the graph of $V_i(\theta; v)$ is above the straight line joining the points $(0, V_i(0; v))$ and $(1, V_i(1; v))$, as shown in Fig. 3.2. So when $v = 0$, $D_i(0) = \varnothing$.

On the other hand, we know that $V_i(\theta, v)$ is upper bounded by $\frac{-\log(0.5)}{1-\beta}$, which is the discounted total cost when the state stays as $\theta = 0.5$, which occurs if no probing occurs. As a result, when $v > \frac{-\log(0.5)}{1-\beta}$, $D_i(v) = \{\theta_i : \theta_i \in [0, 1]\}$.

Next we prove the monotonicity of $D_i(v)$. We know that the LHS of Equation (3.7) is a concave function in $\theta_i$. So the LHS and RHS can be plotted as in Fig. 3.2. From the symmetry of the cost function, we know that $V_i(\theta_i; v) = V_i(1 - \theta_i; v)$, so in

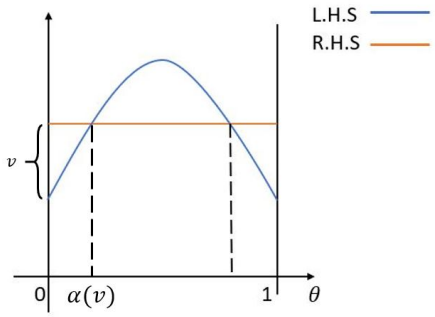



**Figure 3.2:** LHS and RHS of Eq.(3.7)

the remaining part of the proof, we assume that $\theta_i \in [0, 0.5]$ for simplicity.

According to Fig. 3.2, when $\theta_i \in [0, \alpha(v))$, the LHS is smaller than the RHS of Equation (3.7), $D_i(v) = \{[0, \alpha(v))\}$. Let

$$g_1(\theta_i(t), v) = \beta r V_i\big(p_i + (1 - 2p_i)\theta_i(t); v\big)$$
$$g_2(\theta_i(t), v) = v + \beta r V_i(0; v).$$

We can prove the indexability by proving

$$\frac{\partial g_2}{\partial v} - \frac{\partial g_1}{\partial v} \geq 0$$

for any $\theta_i \in [0, \alpha(v))$. In other words, for fixed $\theta$, as $v$ increases, $g_2$ increases faster than $g_1$ if $\theta \in D(v)$. The condition can be written as:

$$1 + \beta r \frac{\partial V_i(0; v)}{\partial v} - \beta r \frac{\partial V_i(p_i + (1 - 2p_i)\theta_i; v)}{\partial v} \geq 0 \tag{3.8}$$

for any $\theta_i \in D_i(v)$.

Here we point out that $V(\theta; v)$ is not differentiable for some $v$. In particular, the function is not differentiable when

$$\beta r V_i\big(p_i + (1-2p_i)\theta; v\big) = v + \beta r V_i(0; v),$$

i.e. when $\theta$ is on the boundary of $D(v)$. When $V_i(\theta; v) = g_1(\theta, v) = g_2(\theta, v)$, probe or not does not make any difference, but the derivative of the two terms in the Bellman equation Equation (3.5) may be different. Since a boundary point is not included in $D(v)$ According to its definition, so we consider the derivative of the second term when it is not differentiable and defines it to be $\frac{\partial V(\theta;v)}{\partial v}$ here, because if Eq.(3.8) holds for all the differentiable points, it also holds for both left and right hand derivative at the non-differentiable points. For the non-differentiable point of $V(0; v)$, right hand derivative will be considered.

Let

$$h^0(\theta) = \theta,$$

and

$$h^t(\theta) = p_i + (1-2p_i)h^{t-1}(\theta)$$

for $t \geq 1$, represents the $t$ step state transition without probe. For any $\theta \in D_i(v)$, let

$$k = \arg\max_k \{h^k(\theta) \in D_i(v)\}.$$

So for $\theta \in D_i(v)$ we have:

$$V_i(g^t(\theta); v) = \sum_{i=0}^{t} c_i(h^i(\theta)) + \beta^{t+1} V_i(h^{t+1}(\theta); v) \tag{3.9}$$

when $0 \leq t \leq k$. The costs $c_i(\theta_i)$ are independent with $v$, so we have:

$$\frac{\partial V_i(\theta; v)}{\partial v} = \beta^{k+1} \frac{\partial V_i(h^{k+1}(\theta); v)}{\partial v} \tag{3.10}$$

As a complement, we also point out that $k$ is an integer related to $v$, $\frac{\partial V_i(h^{k+1}(\theta);v)}{\partial v}$ is not differentiable when $h^{k+1}(\theta)$ lies on the boundary of $D(v)$ for any $k \geq 0$. We will prove that the indexability holds for any $k \geq 0$, then both left-hand derivative and right-hand derivative are under consideration. So we simply regard $k$ as a constant for all differentiable $v$.

Next we consider about the term $\frac{\partial V_i(h^{k+1}(\theta);v)}{\partial v}$, we have:

$$\begin{aligned} V_i(g^{k+t}(\theta);v) = v + c_i(g^{k+t}(\theta)) + \\ \beta r V_i(0;v) + \beta(1-r)V(g^{k+t+1}(\theta);v) \end{aligned} \tag{3.11}$$

for any $t \geq 1$. Let $\frac{\partial V(0;v)}{\partial v} = x$

$$\begin{aligned} \frac{\partial V_i(g^{k+1}(\theta);v)}{\partial v} &= 1 + \beta r x + \beta(1-r) \\ &\Big\{1 + \beta r x + \beta(1-r)\Big(1 + \beta r x + \beta(1-r)\cdots\Big)\Big\} \\ &= \frac{1+\beta r x}{1-\beta(1-r)} \end{aligned}$$

so for $\theta \in D(v)$, we have:

$$\frac{\partial V_i(\theta;v)}{\partial v} = \beta^{k+1}\frac{1+\beta r x}{1-\beta(1-r)}$$

and

$$\frac{\partial V_i(p_i(1-2p_i)\theta_i(t);v)}{\partial v} = \beta^{k}\frac{1+\beta r x}{1-\beta(1-r)}$$

Then Eq.(3.8) becomes:

$$\begin{aligned} 1 + \beta r\left[x - \beta^k \frac{1+\beta r x}{1-(1-r)\beta}\right] &\geq 0 \\ x - \beta^k \frac{1+\beta r x}{1-(1-r)\beta} &\geq -\frac{1}{\beta r} \\ \left(1 - \frac{\beta^{k+1} r}{1-\beta+\beta r}\right) x &\geq \frac{\beta^k}{1-\beta+\beta r} - \frac{1}{\beta r} \end{aligned}$$

First it is easy to show that $\frac{\partial V(\theta;v)}{\partial v} > 0$, since as the penalty increase, the discounted total cost can not decrease, so we have $x \geq 0$, and the LHS of above equation is always positive for any $k \geq 0$.

Since $\frac{\beta^k}{1-\beta+\beta r} < \frac{1}{\beta r}$, the RHS of the above equation is always smaller than 0 for any $k \geq 0$. So Eq.(3.8) always holds when $\theta_i(t) \in D_i(v)$, and the problem is indexable.

As we mentioned before, $V_i(h^{k+1}(\theta), v)$ is not differentiable in $v$ when $h^{k+1}(\theta)$ is on the boundary of $D(v)$, since $k$ is a piece-wise constant on $v$. Both of left derivative and right derivative can be support by Eq.(3.8), since it holds for any $k \geq 0$. □

The value $v^*(\theta_i, r)$ is defined as the penalty on probing to balance the two terms in the Bellman equation Eq.(3.5). Whittle's index based policies are known to have good performance in practice, see Aalto *et al.* (2016) and Anand and de Veciana (2018).

We next summarize the calculation of Whittle's index. According to Eq.(3.9) and Eq.(3.11), $\theta$ is on the boundary of $D(v^*(\theta))$, in other words, $D(v^*(\theta)) = [0, \theta)$ for any $\theta > 0$. So we can get:

$$\begin{aligned} V_i(0, v^*(\theta)) &= \sum_{j=0}^{L_0} \beta^j c_i(h^j(0)) + \beta^{L_0} \Big( v + \\ &\quad \beta \Big[ r V_i(0; v^*(\theta)) + (1-r) V_i\left(h^{L_0+2}(0); v^*(\theta)\right) \Big] \Big) \\ &= \sum_{j=0}^{L_0} \beta^j c_i(h^j(0)) + \beta^{L_0} \frac{v + \beta r V(0; v^*(\theta))}{1 - \beta(1-r)} \\ &\quad + \sum_{j=1}^{\infty} \beta^{L_0+j} (1-r)^j c(h^{L_0+j}(0)) \end{aligned} \tag{3.12}$$

where

$$\begin{aligned} L_0 &= \arg\max_k \{h^k(0) \in D(v^*(\theta))\} \\ &= \lceil \log_{1-2p_i}(1-2\theta) \rceil. \end{aligned}$$

On the other hand, Eq.(3.7) holds for $v = v^*(\theta)$, we have

$$\beta r V_i\big(h^1(\theta); v^*(\theta)\big) = v + \beta r \cdot V_i(0; v^*(\theta))$$

$$\sum_{j=1}^{\infty} r\beta^j(1-r)^{j-1}c(h^j(\theta))+\frac{\beta r v+\beta^2 r^2 V_i(0;v^*(\theta))}{1-\beta(1-r)} \\ = v+\beta r\cdot V_i(0;v^*(\theta)) \tag{3.13}$$

Combine Eq.(3.12) and Eq.(3.13), Whittle's index $v^*(\theta)$ of the $i$th UAV at state $\theta$ can be solved.

### *3.2.2 Multi-State Channel*

We now consider multi-state channel case, assume that the channel states of the $i$th Markov chain $r_i$ is an i.i.d. random variable such that $r_i \in \mathcal{R}_i = \{r_{i,1}, r_{i,2}, \cdots, r_{i,n}\}$ with $r_{i,1} > r_{i,2} > \cdots > r_{i,n}$ for any $i$. Each channel state occurs with probabilities $\rho_{i,1}, \rho_{i,2}, \cdots, \rho_{i,n}$ respectively, and satisfying $\sum_j \rho_{i,j} = 1$ for any $i$. Also we assume that the channel states at current time is known for all Markov chains, but the future channel states are unknown. This setting is similar to the multi-state channel in Anand and de Veciana (2018).

In Multi-State channel, for the $i$th Markov chain, the tuple $(\theta_i, r_i)$ where $\theta_i \in [0,1]$, and $r_i \in \mathcal{R}_i$ consists the state, since decision depends on both $\theta_i$ and $r_i$, and the state space is $[0,1] \times \mathcal{R}_i$. Still, let $D_i(v)$ be the set of states where the $i$th Markov chain would not to be probed under $v$-penalty policy. The Bellman Equation (3.5) becomes:

$$V_i(\theta_i(t), r_i; v) = \min\Big\{c_i(\theta_i(t))+\beta\overline{V}_i\big(p_i+(1-2p_i)\theta_i(t);v\big), \\ v + c_i(\theta_i(t)) + \beta r_i \overline{V}_i(0;v) + \beta(1-r_i)\cdot \\ \overline{V}_i\left(p_i+(1-2p_i)\theta_i(t);v\right)\Big\} \tag{3.14}$$

where $\overline{V}_i(\theta_i; v) = \mathbb{E}_{r_i}[V_i(\theta_i, r_i; v)]$ is the expected value over $r_i$, that is:

$$\overline{V}_i(\theta_i(t);v) = \mathbb{E}_{r_i}\Big[\min\Big\{c_i(\theta_i(t))+\beta\overline{V}_i\big(p_i+(1-2p_i)\theta_i(t);v\big), \\ v+\beta r_i\overline{V}_i(0;v)+\beta(1-r_i)\overline{V}_i\left(p_i+(1-2p_i)\theta_i(t);v\right)\Big\}\Big] \tag{3.15}$$

Similarly, because of the symmetric property of the Markov process, we only consider about $\theta \in [0, 0.5]$. From the previous section, we know that the concavity still holds. Let the two terms in the minimum are equal to each other, we have:

$$\beta r_i \overline{V}_i\big(p_i + (1-2p_i)\theta_i(t); v\big) = v + \beta r_i \overline{V}_i(0; v) \tag{3.16}$$

For agent $i$, the space of $\theta_i$ can be divided into $n+1$ parts $\{\Phi_{i,l}\}$ where $l = 0, 1, 2, \cdots, n$, and $\Phi_{i,0}$ satisfying:

$$\beta r_{i,j} \overline{V}_i\big(p_i + (1-2p_i)\theta_i; v\big) < v + \beta r_{i,j} \overline{V}_i(0; v) \text{ for all } r_{i,j} \in \mathcal{R}_i.$$

$\Phi_{i,1}$ satisfies:

$$\begin{cases} \beta r_{i,1} \overline{V}_i\big(p_i + (1-2p_i)\theta_i; v\big) \geq v + \beta r_{i,1} \overline{V}_i(0; v) \\ \beta r_{i,j} \overline{V}_i\big(p_i + (1-2p_i)\theta_i; v\big) \\ \qquad\qquad < v + \beta r_{i,j} \overline{V}_i(0; v) \end{cases} \text{for all } j > 1$$

$\Phi_{i,l}$ satisfies:

$$\begin{cases} \beta r_{i,j} \overline{V}_i\big(p_i + (1-2p_i)\theta_i; v\big) \\ \qquad\qquad \geq v + \beta r_{i,j} \overline{V}_i(0; v) \end{cases} \text{for all } j \leq l$$

$$\begin{cases} \beta r_{i,j} \overline{V}_i\big(p_i + (1-2p_i)\theta_i; v\big) \\ \qquad\qquad < v + \beta r_{i,j} \overline{V}_i(0; v) \end{cases} \text{for all } j > l$$

$\Phi_{i,n}$ satisfies:

$$\beta r_{i,j} \overline{V}_i\big(p_i(1-2p_i)\theta_i; v\big) \geq v + \beta r_{i,j} \overline{V}_i(0; v) \text{ for all } r_{i,j} \in \mathcal{R}_i.$$

From the concave property, we have $\overline{V}_i\big(p_i(1-2p_i)\theta_i; v\big) > \overline{V}_i(0; v)$ for any $\theta_i$, by moving the $\beta r_{i,j} \overline{V}(0, v)$ term to the left, it is easy to show that any $\theta_i \in \Phi_{i,l-1}$ is smaller than $\theta_i' \in \Phi_{i,l}$. Then the rested set $D_i(v)$ of the $i$th Markov chain can be described as $D_i(v) = \{(\theta_i, r_i) : \theta_i \in \Phi_{i,0}, r_i \in \mathcal{R}_i \text{ or } \theta_i \in \Phi_{i,l}, r_i < r_{i,l} \text{ for } 0 < l \leq n\}$.

Similarly, to prove the indexability of *iid* situation, we need to prove

$$1 + \beta r_i \frac{\partial \overline{V}_i(0; v)}{\partial v} - \beta r_i \frac{\partial \overline{V}_i(p_i + (1 - 2p_i)\theta_i; v)}{\partial v} \geq 0 \tag{3.17}$$

holds for any $(\theta_i, r_i) \in D_i(v)$ for any $i$.

**Theorem 3** *The mult-state channel Markov chains are indexable when*

$$\beta < \frac{1}{1 + (1 - \rho_{i,1}) r_{i,1}}, \tag{3.18}$$

*holds for all $i$.*

The proof of this Theorem can be found in the appendix.

Similarly, the we would choose to probe the $K$ Markov chains with higher indexes. However the indexes now depend on both state estimation $\theta_i(t)$ and channel state $r_i$. In multi-state channel, the explicit format of $v_i^*(\theta, r_i)$ is hard to solve, especially when the number of state of channel is large. However, we can use binary search to get an approximation. As for an example, we will show brief process to derive the two-state channel index as an example.

For the $i$th Markov chain, to solve the index $v^*(\theta, r_{i,1})$, let $v^* = v^*(\theta, r_{i,1}) = v^*(\theta', r_{i,2})$ $(0 < \theta < \theta' < 0.5)$, and $\theta'$ is temporarily unknown. From the proof above, let $\overline{V}_i(0, v^*) = x$, $(\theta, r_{i,1})$ and $(\theta', r_{i,2})$ is on the boundary of $D_i(v^*)$, we have the following equations:

$$\beta r_{i,1} \overline{V}_i(p_i + (1 - 2p_i)\theta; v^*) = v^* + \beta r_{i,1} x \tag{3.19}$$

$$\beta r_{i,2} \overline{V}_i(p_i + (1 - 2p_i)\theta'; v^*) = v^* + \beta r_{i,2} x \tag{3.20}$$

On the other hand, $x = \overline{V}_i(0, v^*)$ can be expressed as:

$$
\begin{aligned}
x = & \sum_{j=0}^{L_0} \beta^j c_i(h_i^j(0)) + \sum_{j=0}^{L_1-1} \beta^{L_0+1+j}(1-\rho_1 r_1)^j \\
& \Big(c_i(h_i^{L_0+1+j}(0)) + \rho_{i,1} v^* + \beta \rho_{i,1} r_{i,1} x\Big) \mathbb{1}\{L_1 > 0\} + \\
& \sum_{j=0}^{\infty} \beta^{L_0+L_1+1+j}(1-\rho_{i,1} r_{i,1})^{L_1}(1-\overline{r}_i)^j \\
& \Big(c_i(h_i^{L_0+L_1+1+j}(0)) + v^* + \beta \overline{r}_i x\Big)
\end{aligned} \tag{3.21}
$$

where $L_0 = \max_k\{h_i^k(0) < \theta\}$, and $L_1 = \max_k\{h^k(0) < \theta'\} - k_0$. Combine Eq.(3.19) (3.20) (3.21) the index value of $v^*$, $\theta'$, $\overline{V}_i(0, v^*)$ for the $i$th Markov chain can be estimated by using binary search.

## 3.3 Simulations

We consider a scenario where a control tower is monitoring UAVs in the area. Each UAV has two states: "low risk" and "high risk". The transition probability $p$ from one state to another is assumed to be 0.05.

We assume that there are 500 UAVs in the system, based on the channel bandwidth, the control tower can require information from 150 of them at each time slot. Assume that there are two types of UAV, each types has 250 UAVs. The first one has transition probability $p_1 = 0.05$, and has transmission success probability $r_1 = 0.5$. The second type of UAV has transition probability $p_2 = 0.02$ and transmission success probability $r_2 = 0.7$.

We can plot the index of these two types of UAV as in Fig.3.3. We compare the Whittle's index approach with a greedy method that UAVs with larger $c(\theta)$ will be selected, and the Round Robin method, where all the UAVs are selected periodically with same frequency. And the simulation results of total information entropy of all these 500 UAVs are shown in Fig.3.4. As we can see, the information entropy level of

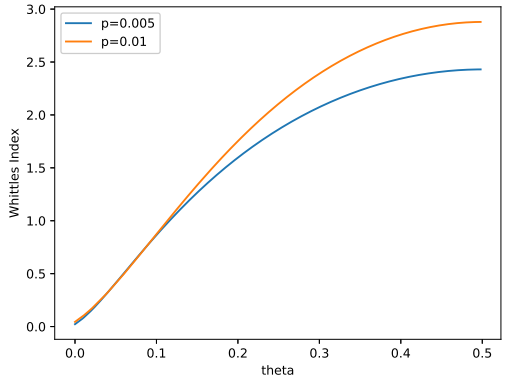



**Figure 3.3:** Index of two types of UAV

Whittle's Index Approach is lower than both the greedy or Round & Robin methods.

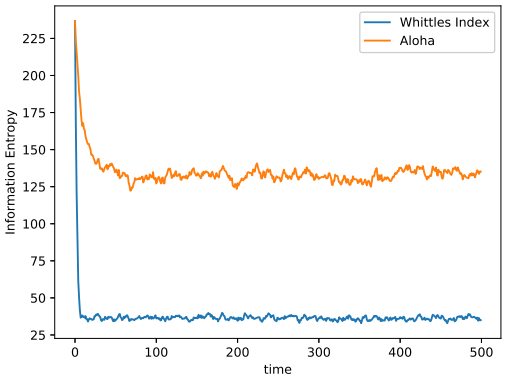



**Figure 3.4:** Information Entropy Simulation

We next consider multi-state channels such that $\Pr(r = 0.9) = 0.4$, $\Pr(r = 0.7) = 0.3$, and $\Pr(r = 0.5) = 0.3$.

The simulation results are plotted in Fig.3.5. Again we can observe that the Whittle's index outperforms the other two algorithms.

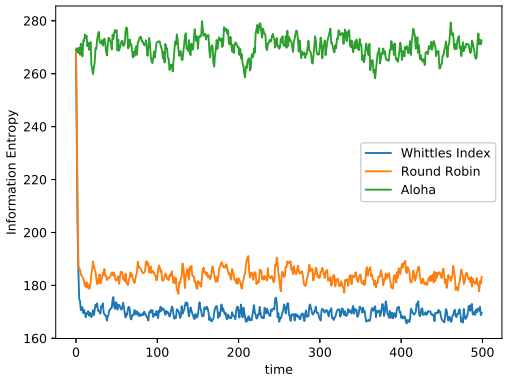



**Figure 3.5:** Index of Three States Channel UAV

# Chapter 4

# CONCLUSION AND FUTURE WORK

In this paper, we proposed our 3M-RL for UAV collision free routing and implemented the algorithm with CNN and Actor-Critic. This method overcomes the curse of dimensionality, and significantly reduces the computational complexity compared with centralized path planning algorithms. 3M-RL takes intruder information with different resolutions based on their distances. For nearby intruders, detailed information is acquired to avoid the collision. For the intruders in a medium range, it focuses on their mean direction. For the intruders far away, the probability of their approaching can also be learned during the training. We constructed two network deep-neural-network structures fits for different sizes of space.

We also studied the problem of learning the states of parallel Markov chains over unreliable wireless networks. The solution to this problem has applications in risk monitoring such as monitoring the states of UAVs from a control tower. We first proved that for single state wireless channels, the problem based on Whittle's index is indexable, and the index can be derived explicitly. For multi-state channels, the indexability can be proved under a sufficient condition, and the index value can be calculated numerically. And simulation shows that the proposed Whittle's index approach can have better performance than greedy policy and Round & Robin policy.

Although we have simulated our 3M-RL algorithm in 3D space, the model we used is still a crude virtual model, this simulation is just used to show the ability of our algorithm in handling observations and collision problems in 3D continuous space. We will consider about more accurate model in the future, e,g, models that controlling pitching angle and reasonable step response. Also, algorithms like Šišlák

*et al.* (2010) considers the spatial-temporal collision avoidance in 4D domain (3-D space and time), we could also introduce time index as part of our state, and extends the policy to be observation-time dependent, which may able to solve the collision avoidance between agents and external moving obstacles.

On the other hand, our algorithm is a data based learning algorithm, the outcome policy depends on the distribution of data we get in the training process. The guarantee of reinforcement learning collision avoidance requires meticulous stability and optimality proof. The proof in multi-agent reinforcement learning algorithms is even more difficult, the researches on the optimality and stability of multi-agent reinforcement learning algorithm is also very constrainedZhang *et al.* (2019b). We consider the stability and optimality proof of our algorithm as a significant future work.

Furthermore, communications among UAV systems usually are unreliable, and usually suffer from delay and loss Qiu *et al.* (2019, 2020). Solving the collision avoidance in wireless communication network could also be an interesting problem to study.

Simulation results showed that our method can provide good paths. The method also has a fast learning rate compared with mean-field Q-learning, and results in a lower total cost compared with the mean-field multi-agent reinforcement learning.

REFERENCES


Aalto, S., P. Lassila and P. Osti, "Whittle index approach to size-aware scheduling for time-varying channels with multiple states", Queueing Systems **83**, 3, 195–225, URL `https://doi.org/10.1007/s11134-016-9484-z` (2016).

Anand, A. and G. de Veciana, "A whittle's index based approach for qoe optimization in wireless networks", Proc. ACM Meas. Anal. Comput. Syst. **2**, 1, 15:1–15:39, URL `http://doi.acm.org/10.1145/3179418` (2018).

Bakule, L., "Decentralized control: An overview", Annual reviews in control **32**, 1, 87–98 (2008).

Carbone, C., U. Ciniglio, F. Corraro and S. Luongo, "A novel 3d geometric algorithm for aircraft autonomous collision avoidance", in "Proceedings of the 45th IEEE Conference on Decision and Control", pp. 1580–1585 (2006).

Chen, Y. F., M. Liu, M. Everett and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning", CoRR **abs/1609.07845**, URL `http://arxiv.org/abs/1609.07845` (2016).

Christodoulou, M. A. and S. G. Kodaxakis, "Automatic commercial aircraft-collision avoidance in free flight: the three-dimensional problem", IEEE Transactions on Intelligent Transportation Systems **7**, 2, 242–249 (2006).

Christodoulou, P., "Soft actor-critic for discrete action settings", arXiv preprint arXiv:1910.07207 (2019).

Ferrera, E., A. Alcántara, J. Capitán, A. R. Castaño, P. J. Marrón and A. Ollero, "Decentralized 3d collision avoidance for multiple uavs in outdoor environments", Sensors **18**, 12, 4101 (2018).

Galati, G., E. G. Piracci, N. Petrochilos and F. Fiori, "1090 mhz channel capacity improvement in the air traffic control context", in "2008 Tyrrhenian International Workshop on Digital Communications - Enhanced Surveillance of Aircraft and Vehicles", pp. 1–5 (2008).

Hu, J., X. Yang, W. Wang, P. Wei, L. Ying and Y. Liu, "Uas conflict resolution in continuous action space using deep reinforcement learning", in "AIAA AVIATION 2020 FORUM", p. 2909 (2020).

Inc, T. M., "Uav toolbox user's guide", **1.0 (R2020b)**, URL `https://www.mathworks.com/help/pdf`$_{d}oc/uav/uav_{u}g.pdf$(2020).

Jenie, Y. I., E. van Kampen, J. Ellerbroek and J. M. Hoekstra, "Safety assessment of a uav cd r system in high density airspace using monte carlo simulations", IEEE Transactions on Intelligent Transportation Systems **19**, 8, 2686–2695 (2018).

Julian, K., M. Kochenderfer and M. Owen, "Deep neural network compression for aircraft collision avoidance systems", Journal of Guidance, Control, and Dynamics **42**, 3, 598–608, URL

Kadanoff, L. P., "More is the same; phase transitions and mean field theories", Journal of Statistical Physics **137**, 5-6, 777 (2009).

Lin, Y. and S. Saripalli, "Sampling-based path planning for uav collision avoidance", IEEE Transactions on Intelligent Transportation Systems **18**, 11, 3179–3192 (2017).

Lin, Z., L. Castano, E. Mortimer and H. Xu, "Fast 3d collision avoidance algorithm for fixed wing uas", Journal of Intelligent & Robotic Systems **97**, 3, 577–604 (2020).

Liu, H., X. Li, M. Fan, G. Wu, W. Pedrycz and P. N. Suganthan, "An autonomous path planning method for unmanned aerial vehicle based on a tangent intersection and target guidance strategy", IEEE Transactions on Intelligent Transportation Systems pp. 1–13 (2020).

Liu, K. and Q. Zhao, "Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access", IEEE Transactions on Information Theory **56**, 11, 5547–5567 (2010).

Long, P., T. Fan, X. Liao, W. Liu, H. Zhang and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning", in "2018 IEEE International Conference on Robotics and Automation (ICRA)", pp. 6252–6259 (IEEE, 2018).

Lowe, R., Y. Wu, A. Tamar, J. Harb, P. Abbeel and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments", CoRR **abs/1706.02275**, URL `http://arxiv.org/abs/1706.02275` (2017).

Maturana, D. and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition", in "2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)", pp. 922–928 (2015).

Maza, I., K. Kondak, M. Bernard and A. Ollero, "Multi-uav cooperation and control for load transportation and deployment", in "Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009", pp. 417–449 (Springer, 2009).

Menouar, H., I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri and A. Tuncer, "Uav-enabled intelligent transportation systems for the smart city: Applications and challenges", IEEE Communications Magazine **55**, 3, 22–28 (2017).

Pallottino, L., E. M. Feron and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming", IEEE Transactions on Intelligent Transportation Systems **3**, 1, 3–11 (2002).

Pinto, L., M. Andrychowicz, P. Welinder, W. Zaremba and P. Abbeel, "Asymmetric actor critic for image-based robot learning", arXiv preprint arXiv:1710.06542 (2017).

Qiu, J., D. Grace, G. Ding, J. Yao and Q. Wu, "Blockchain-based secure spectrum trading for unmanned-aerial-vehicle-assisted cellular networks: An operator's perspective", IEEE Internet of Things Journal **7**, 1, 451–466 (2020).

Qiu, J., D. Grace, G. Ding, M. D. Zakaria and Q. Wu, "Air-ground heterogeneous networks for 5g and beyond via integrating high and low altitude platforms", IEEE Wireless Communications **26**, 6, 140–148 (2019).

Qu, G., A. Wierman and N. Li, "Scalable reinforcement learning of localized policies for multi-agent networked systems", (2020).

Raghunathan, A. U., V. Gopal, D. Subramanian, L. T. Biegler and T. Samad, "Dynamic optimization strategies for three-dimensional conflict resolution of multiple aircraft", Journal of guidance, control, and dynamics **27**, 4, 586–594 (2004).

Richards, A. and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming", in "Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)", vol. 3, pp. 1936–1941 vol.3 (2002).

Singla, A., S. Padakandla and S. Bhatnagar, "Memory-based deep reinforcement learning for obstacle avoidance in UAV with limited environment knowledge", CoRR **abs/1811.03307**, URL `http://arxiv.org/abs/1811.03307` (2018).

Šišlák, D., P. Volf and M. Pěchouček, "Agent-based cooperative decentralized airplane-collision avoidance", IEEE Transactions on Intelligent Transportation Systems **12**, 1, 36–46 (2010).

Subramanian, J. and A. Mahajan, "Reinforcement learning in stationary mean-field games", in "Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems", AAMAS '19, p. 251–259 (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2019).

Sutton, R. S. and A. G. Barto, *Reinforcement learning: An introduction* (MIT press, 2018).

van den Berg, J., S. J. Guy, M. Lin and D. Manocha, "Reciprocal n-body collision avoidance", in "Robotics Research", edited by C. Pradalier, R. Siegwart and G. Hirzinger, pp. 3–19 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2011).

Whittle, P., "Restless bandits: activity allocation in a changing world", Journal of Applied Probability **25**, A, 287–298 (1988).

Yang, J., D. Yin, Q. Cheng and L. Shen, "Two-layered mechanism of online unmanned aerial vehicles conflict detection and resolution", IEEE Transactions on Intelligent Transportation Systems **19**, 10, 3230–3244 (2018).

Yang, Y., R. Luo, M. Li, M. Zhou, W. Zhang and J. Wang, "Mean field multi-agent reinforcement learning", (2018).

Young, H. P., "The evolution of conventions", Econometrica **61**, 1, 57–84, URL `http://www.jstor.org/stable/2951778` (1993).

Zhang, K., Z. Yang and T. Basar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms", CoRR **abs/1911.10635**, URL `http://arxiv.org/abs/1911.10635` (2019a).

Zhang, K., Z. Yang and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms", arXiv preprint arXiv:1911.10635 (2019b).

Zhang, Z., P. Jaiswal and R. Rai, "Featurenet: Machining feature recognition based on 3d convolution neural network", Computer-Aided Design **101**, 12–22 (2018).

Zhao, P., W. Wang, L. Ying, B. Sridhar and Y. Liu, "Online multiple-aircraft collision avoidance method", Journal of Guidance, Control, and Dynamics **43**, 8, 1456–1472 (2020).

Zhu, L., X. Cheng and F.-G. Yuan, "A 3d collision avoidance strategy for uav with physical constraints", Measurement **77**, 40–49 (2016).

# APPENDIX A

# PROOF

Proof of Lemma 1

**Proof:** We start with a finite time problem, let $T$ be the time horizon, and $V_i^*(\theta_i(t), t; v)$ be the minimum discounted cost start from state $\theta_i(t)$ at time $t$, and the process will be terminated at time $T$. We have

$$V_i^*(\theta_i(t), t; v) = \min_{\pi} \Big\{ \sum_{\tau=t}^{T-1} \beta^{\tau-t}[c_i(\theta_i(\tau)) + v\mathbb{1}_{A_i^{\pi}(\tau)=1}] + \beta^{T-t} V_i(\theta_i(T), T; v) \Big\} \tag{A.1}$$

is the cost from time $t$ to time horizon $T$ under policy $\pi$. We assume that the terminal cost $V_i(\theta_i(T), T; v)$ is always zero. When $t = T-1$, we have:

$$V_i(\theta_i(T-1), T-1; v) = \min\{c_i(\theta_i(T-1)), v + c_i(\theta_i(T-1))\}$$

It is obviously that

$$\begin{aligned} c_i(\theta_i(T-1)) = &- \theta_i(T-1)\log(\theta_i(T-1)) - (1-\theta_i(T-1)) \\ &\log(1-\theta_i(T-1)) \end{aligned}$$

is a concave function on $\theta_i(T-1)$, and $v$ is a constant. Since the point-wise minimum of two concave functions is still concave, we can get $V_i(\theta_i(T-1), T-1; v)$ is concave on $\theta_i(T-1)$.
For any $t < T-1$, if $V_i(\theta_i(t+1), t+1; v)$ is concave on $\theta_i(t+1)$, we have:

$$\begin{aligned} V_i(\theta_i(t),t;v) = \min\Big\{ & c_i(\theta_i(t)) + \beta V_i\big(p_i + (1-2p_i)\theta_i(t), t+1; v\big), \\ & v + c_i(\theta_i(t)) + \beta\Big[ r\theta_i(t) V_i(1, t+1; v) + r(1-\theta_i(t))\cdot \\ & V_i(0, t+1; v) + (1-r) V_i\left(p_i + (1-2p_i)\theta_i(t), t+1; v\right)\Big]\Big\} \end{aligned} \tag{A.2}$$

The first term in the minimum is concave since it is the sum of two concave functions. For the second term, because of the symmetric of the cost function, $V_i(1, t+1; v) = V_i(0, t+1; v)$ are constants, the second term is also concave since it is summation of constants and concave functions. As a result, $V_i(\theta_i(t),t;v)$ is also concave.
As $T \to \infty$, for any finite $t$, $V_i(\theta_i(t),t;v)$ can be regard as point-wise minimum of infinite concave functions. According to the property of epigraph, and the fact that convexity is preserved under intersection, we can have:

$$\lim_{T\to\infty} V_i^*(\theta_i(t), t; v) = V_i^*(\theta_i(t); v)$$

is a concave function on $\theta_i(t)$. □

Proof of Theorem 3

**Proof:** Similarly, we point out that $\overline{V}_i(\theta; v)$ is not differentiable on $v$ for some $v$, for example, when $\theta$ is on the boundary of $D_i(v)$. According to the definition of $D_i(v)$, the point on the boundary is not included in $D_i(v)$, so we consider the derivative of the second term when it is not differentiable. We just define $\frac{\partial \overline{V}_i(\theta;v)}{\partial v}$ here, because if Eq.(3.17) holds for all the differentiable points, it also holds for both left and right hand derivative at the non-differentiable points.
We start from $\theta \in \Phi_{i,0}$, we need to prove that Eq.(3.17) holds for any $\theta \in \Phi_{i,0}$ and $r_i \in \mathcal{R}_i$. Similarly, let

$$h_i^k(\theta) = \frac{1-(1-2p_i)^k}{2} + (1-2p_i)^k\theta$$

is $k$-step transition without probing. Let

$$k_{i,0} = \arg\max_k \{h_i^k(\theta) \in \Phi_{i,0}\},$$

we have $k_{i,0} \geq 0$, since

$$\beta r_i \overline{V}_i\big(p_i + (1-2p_i)\theta; v\big) < v + \beta r_i \overline{V}_i(0; v)$$

for $\theta \in \Phi_{i,0}$. Then we have:

$$\overline{V}_i(\theta, v) = \sum_{j=0}^{k_{i,0}} \beta^j c(h_i^j(\theta)) + \beta^{k_{i,0}+1} \overline{V}_i(h_i^{k_{i,0}+1}(\theta), v).$$

and we have:

$$\frac{\partial \overline{V}_i(\theta, v)}{\partial v} = \beta^{k_{i,0}+1} \frac{\partial \overline{V}_i(h_i^{k_{i,0}+1}(\theta), v)}{\partial v}.$$

Similarly, $\frac{\partial \overline{V}_i(h_i^{k_{i,0}+1}(\theta), v)}{\partial v}$ is not differentiable in $v$ when $h_i^{k_i+1}(\theta)$ lies on the boundary of $D_i(v)$ for any $k_i \geq 0$. We will propose a sufficient condition for the indexability in for any $k_{i,0} \geq 0$, such that both left derivative and right derivative. So we simply regard $k_0$ as a constant for all differentiable $v$. Then let $k_{i,1} = \max_k \{h_i^{k_{i,0}+k}(\theta) \in \Phi_{i,0} \cup \Phi_{i,1}\}$, $k_{i,1} \geq 0$ since $h_i^{k_{i,0}}(\theta) \in \Phi_{i,0}$, we have:

$$\begin{aligned}\overline{V}_i(h_i^{k_{i,0}+1}(\theta); v) = c(h_i^{k_{i,0}+1}(\theta)) + \beta(1-\rho_{i,1}r_{i,1})\cdot \\ \overline{V}(h_i^{k_{i,0}+2}(\theta); v) + \rho_1 v + \beta\rho_1 r_1 \overline{V}_i(0; v)\end{aligned}$$

And assume $\frac{\partial \overline{V}_i(p_i;0)}{\partial v} = x$,

$$\begin{aligned}\frac{\partial \overline{V}_i(h_i^{k_{i,0}+1}(\theta);v)}{\partial v} =& \rho_{i,1} + \rho_{i,1}\beta r_{i,1}x + \\ & \beta(1-\rho_{i,1}r_{i,1})\frac{\partial \overline{V}_i(h_i^{k_{i,0}+2}(\theta);v)}{\partial v} \\ =& (\rho_{i,1}+\rho_{i,1}\beta r_{i,1}x)\frac{1-\beta^{k_{i,1}}(1-\rho_{i,1}r_{i,1})^{k_{i,1}}}{1-\beta(1-\rho_{i,1}r_{i,1})} + \\ & \beta^{k_{i,1}}(1-\rho_{i,1}r_{i,1})^{k_{i,1}}\frac{\partial \overline{V}_i(h_i^{k_{i,0}+k_{i,1}+1}(\theta);v)}{\partial v}\end{aligned}$$

Then let $k_{i,2} = \max_k\{h_i^{k_{i,0}+k_{i,1}+k}(\theta) \in \Phi_{i,0} \cup \Phi_{i,1} \cup \Phi_{i,2}\}$, similarly, it is easy to show that $k_{i,2} \geq 0$, we have:

$$\begin{aligned}&\overline{V}_i(h_i^{k_{i,0}+k_{i,1}+1}(\theta);v) = c(h_i^{k_{i,0}+k_{i,1}+1}(\theta)) + \\ &\beta(1-\rho_{i,1}-\rho_{i,2})\overline{V}(h_i^{k_{i,0}+k_{i,1}+2}(\theta);v)) + \\ &\rho_{i,1}\Big(v+\beta r_{i,1}\overline{V}_i(0;v)+\beta(1-r_{i,1})\overline{V}_i\left(h_i^{k_{i,0}+k_{i,1}+2}(\theta);v\right)\Big) + \\ &\rho_{i,2}\Big(v+\beta r_{i,2}\overline{V}_i(0;v)+\beta(1-r_{i,2})\overline{V}_i\left(h_i^{k_{i,0}+k_{i,1}+2}(\theta);v\right)\Big)\end{aligned}$$

And still assume $\frac{\partial \overline{V}_i(p_i;0)}{\partial v} = x$,

$$\begin{aligned}&\frac{\partial \overline{V}_i(h_i^{k_{i,0}+k_{i,1}+1}(\theta);v)}{\partial v} = \rho_{i,1}+\rho_{i,2}+\beta(\rho_{i,1}r_{i,1}+\rho_{i,2}r_{i,2})x + \\ &\qquad \beta(1-\rho_{i,1}r_{i,1}-\rho_{i,2}r_{i,2})\frac{\partial \overline{V}_i(h_i^{k_{i,0}+k_{i,1}+2}(\theta);v)}{\partial v} \\ &= (\rho_{i,1}+\rho_{i,2}+\beta(\rho_{i,1}r_{i,1}+\rho_{i,2}r_{i,2})x)\cdot \\ &\frac{1-\beta^{k_{i,2}}(1-\rho_{i,1}r_{i,1}-\rho_{i,2}r_{i,2})^{k_{i,2}}}{1-\beta(1-\rho_{i,1}r_{i,1}-\rho_{i,2}r_{i,2})} + \\ &\beta^{k_{i,2}}(1-\rho_{i,1}r_{i,1}-\rho_{i,2}r_{i,2})^{k_{i,2}}\frac{\partial \overline{V}_i(h_i^{k_{i,0}+k_{i,1}+1}(\theta);v)}{\partial v}\end{aligned}$$

And until $k_{i,n} = \max_k\{h_i^{\sum_{j=0}^{n-1}k_{i,j}+k}(\theta) \in \cup_{j=0}^{n}\Phi_{i,j}\}$, we have:

$$\begin{aligned}&\overline{V}_i(h_i^{\sum_{j=1}^{n}k_{i,j}+1}(\theta);v) = c_i(h^{\sum_{j=1}^{n}k_{i,j}+1}(\theta)) + \\ &\sum_{j=1}^{n}\rho_{i,j}\Big(v+\beta r_{i,j}\overline{V}_i(0;v)+\beta(1-r_{i,j})\overline{V}_i\left(h^{\sum_{j=1}^{n}k_{i,j}+2}(\theta);v\right)\Big)\end{aligned}$$

and we can have:

$$\frac{\partial \overline{V}_i(h_i^{\sum_{j=1}^{n}k_{i,j}+1}(\theta);v)}{\partial v} = (1+\beta\overline{r}_i x)\frac{1}{1-\beta(1-\overline{r}_i)}$$

where $\overline{r}_i = \sum_{j=0}^{n} \rho_{i,j} r_{i,j}$. In conclusion, each equation can be seen as a the sum of geometric progression, such that they can be written as:

$$\begin{aligned}\frac{\partial \overline{V}_i(h_i^{\sum_{j=0}^{m-1} k_{i,j}+1}(\theta); v)}{\partial v} =& a_{i,m} + a_{i,m} q_{i,m} + \cdots + a_{i,m} q_{i,m}^{k_{i,m}-1} \\ &+ q_{i,m}^{k_{i,m}} \frac{\partial \overline{V}_i(h_i^{\sum_{j=0}^{m} k_{i,j}+1}(\theta); v)}{\partial v}\end{aligned}$$

where $a_{i,m} = (\sum_{j=1}^{m} \rho_{i,j}) + (\beta \sum_{j=1}^{m} r_{i,j} \rho_{i,j})x$, and $q_{i,m} = \beta(1 - \sum_{j=1}^{m} r_{i,j} \rho_{i,j})$. Let $a_i = \max_m\{a_{i,m}\} = 1 + \beta \overline{r}_i x$, and $q_i = \max_m\{q_{i,m}\} = \beta(1 - p_{i,1} r_{i,1})$. So we have:

$$\begin{aligned}\frac{\partial \overline{V}_i(h_i^{k_{i,0}+1}(\theta); v)}{\partial v} &\leq \frac{a_i}{1 - q_i} \\ &= (1 + \beta \overline{r}_i x) \frac{1}{1 - \beta(1 - \rho_{i,1} r_{i,1})}\end{aligned}$$

and

$$\begin{aligned}\frac{\partial \overline{V}_i(\theta, v)}{\partial v} &\leq \beta^{k_{i,0}+1}(1 + \beta \overline{r}_i x) \frac{1}{1 - \beta(1 - \rho_{i,1} r_{i,1})} \\ \frac{\partial \overline{V}_i(p_i + (1 - 2p_i)\theta, v)}{\partial v} &\leq \beta^{k_{i,0}}(1 + \beta \overline{r}_i x) \frac{1}{1 - \beta(1 - \rho_{i,1} r_{i,1})}\end{aligned}$$

So for $\theta \in \Phi_{i,0}$, if we need to prove:

$$\begin{aligned}& 1 + \beta r_i \frac{\partial \overline{V}_i(0; v)}{\partial v} - \beta r_i \frac{\partial \overline{V}_i(p_i + (1 - 2p_i)\theta_i; v)}{\partial v} \\ & \geq 1 + \beta r_i \left( x - \beta^{k_{i,0}} \frac{1 + \beta \overline{r}_i x}{1 - \beta(1 - \rho_1 r_1)} \right) \\ & \geq 0\end{aligned}$$

holds for all $r_i \in \mathcal{R}_i$, we need to have:

$$\begin{aligned}& 1 + \beta r_i \left( x - \beta^{k_{i,0}} \frac{1 + \beta \overline{r}_i x}{1 - \beta(1 - \rho_{i,1} r_{i,1})} \right) \geq 0 \\ & \left( 1 - \frac{\beta^{k_{i,0}+1} \overline{r}_i}{1 - \beta + \beta \rho_{i,1} r_{i,1}} \right) x \geq \frac{\beta^{k_{i,0}}}{1 - \beta + \beta \rho_{i,1} r_{i,1}} - \frac{1}{\beta r_i}\end{aligned}$$

If

$$\beta < \frac{1}{1 + (1 - \rho_{i,1}) r_{i,1}}, \tag{A.3}$$

we have $1 - \beta + \beta \rho_{i,1} r_{i,1} \geq \beta r_{i,1}$, the RHS of the above equation is always negative, the LHS is always positive, and the condition holds for all $k_{i,0} \geq 0$.
Next we consider $\theta \in \Phi_{i,1}$, similarly, still $k_{i,1} = \max_k\{h_i^k(\theta) \in \Phi_{i,0} \cup \Phi_{i,1}\}$, we have:

$$\begin{aligned}\overline{V}_i(h_i^j(\theta); v) =& c(h_i^j(\theta)) + \beta(1 - \rho_{i,1} r_{i,1}) \overline{V}(h_i^{j+1}(\theta); v) + \rho_{i,1} v + \\ & \beta \rho_{i,1} r_{i,1} \overline{V}_i(0; v)\end{aligned}$$

for all $0 \le j \le k_{i,1}$, and

$$\frac{\partial \overline{V}_i(\theta; v)}{\partial v} = (\rho_{i,1} + \rho_{i,1}\beta r_{i,1} x)\frac{1 - \beta^{k_1}(1 - \rho_{i,1} r_{i,1})^{k_{i,1}}}{1 - \beta(1 - \rho_{i,1} r_{i,1})} + \beta^{k_{i,1}}(1 - \rho_{i,1} r_{i,1})^{k_{i,1}} \frac{\partial \overline{V}_i(h_i^{k_{i,1}+1}(\theta); v)}{\partial v}$$

The following parts are same as $\theta \in \Phi_{i,0}$ case. At last, we can also have:

$$\frac{\partial \overline{V}_i(h^{k_{i,0}+1}(\theta); v)}{\partial v} \le (1 + \beta \overline{r}_i x)\frac{1}{1 - \beta(1 - \rho_{i,1} r_{i,1})}$$

For $\theta \in \Phi_{i,1}$, if we need to prove:

$$\begin{aligned} & 1 + \beta r \frac{\partial \overline{V}_i(0; v)}{\partial v} - \beta r \frac{\partial \overline{V}_i(p_i + (1 - 2p_i)\theta; v)}{\partial v} \\ & \ge 1 + \beta r_i \left( x - \frac{1 + \beta \overline{r}_i x}{1 - \beta(1 - \rho_{i,1} r_{i,1})} \right) \\ & \ge 0 \end{aligned}$$

holds for all $r_i < r_{i,1}$. Under the condition of Eq.(3.18), it is easy to prove that it holds for all $k_{i,1} \ge 0$.
Similarly, if $\theta \in \Phi_{i,l}$, we will have $a_i = \max\{a_{i,m}, m = l, l+1, l+2, \cdots, n\} = 1 + \beta \overline{r}_i x$, and $q_i = \max\{q_{i,m}, m = l, l+1, l+2, \cdots, n\} = \beta(1 - \sum_{j=0}^{l} \rho_{i,j} r_{i,j})$. Then the indexability condition Eq.(3.16) becomes:

$$\begin{aligned} & 1 + \beta r_i \frac{\partial \overline{V}_i(p_i; v)}{\partial v} - \beta r_i \frac{\partial \overline{V}_i(p_i + (1 - 2p_i)\theta_i; v)}{\partial v} \\ & \ge 1 + \beta r_i \left( x - \frac{1 + \beta \overline{r}_i x}{1 - \beta(1 - \sum_{j=0}^{l} p_{i,j} r_{i,j}))} \right) \\ & \ge 1 + \beta r_i \left( x - \frac{1 + \beta \overline{r}_i x}{1 - \beta(1 - p_{i,1} r_{i,1})} \right) \\ & \ge 0 \end{aligned}$$

Holds for all $r_i < r_{i,l}$. Under the sufficient condition Eq.(3.18), the indexability for $\theta \in \Phi_{i,l}$ holds for all $k_{i,l} \ge 0$, the multi-state channel Markov chains can be proved to be indexable. □