MULTI-AGENT RL

CSE 510 Introduction to Reinforcement Learning

Srisai Karthik Neelamraju, 50316785, neelamra@buffalo.edu

Anantha Srinath Sedimbi, 50315869, asedimbi@buffalo.edu

06-May-2020

University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

Agenda

- Motivation
- Environment
- Agent Description
- Observations
- Results
- Future Scope

Motivation

- Autonomous agents can be of great help to humans in a variety of hazardous environments.
- Examples: Fire outbreak, natural disasters, rescue operations, etc.
- Though humans can handle some of them, the risk of casualties is high.
- Example: Australian bushfires of 2019-20 have claimed 46M acres of land and lives of 34 personnel.
- Employing single autonomous agent can be of help; but may not be adequate.
- We need multiple agents working towards a common goal (MARL).
- Fire outbreaks are a common hazard across the globe. They have been claiming lives despite sophisticated machinery, trained personnel and several automatic alerting systems. Hence, we want to tackle this problem with the help of MARL.

Problem Definition

- Goals: To define an environment that can
  - host multiple targets and multiple agents
  - host at least 3 different targets
  - host at least 3 different agents
  - support communication among the agents
- An agent should be able to call for help. All other agents should be able to respond.
- Solve the environment using any reinforcement learning algorithm

Related Works:

- Kun Tian & Shuping Jiang (2018) Reinforcement learning for safe evacuation time of fire in Hong Kong-Zhuhai-Macau immersed tube tunnel, Systems Science & Control Engineering, 6:2, 45-56, DOI: 10.1080/21642583.2018.1509746
  - Research focus is to make use of RL Agents for evacuation help in crowded and low-visibility conditions such as tunnels with fire outbreaks.
- Lazaridou, Angeliki, Alexander Peysakhovich, and Marco Baroni. "Multi-agent cooperation and the emergence of (natural) language." arXiv preprint arXiv:1612.07182 (2016).
  - Research focus is make RL agents learn to evolve a ‘Natural Language’ from a vocabulary and to learn communication in that language.
- Leibo, Joel Z., et al. "Multi-agent reinforcement learning in sequential social dilemmas." arXiv preprint arXiv:1702.03037 (2017).
  - Research focus is to establish cooperation between agents to sustain small negative rewards for the sake of a common goal that can be rewarding for all the agents in the environment ultimately.

Environment – "CityOnFire"

- Square Grid World of adaptive sizes – models a city scape
- Each tiny square can be though of as a city block.
- No of fires: min:1, max: no limit defined (50 is a good extreme)
- Each fire is defined as an object with location and heat
- A global counter numbers each fire's ID
- All fires spawn with heat == 100.0
- Fires start at random locations with every episode.

Stochasticity:

- Each fire in the grid will increase its heat stochastically by 2.5 or 5.0 degrees with every step of the environment.
- Initialization is always random.

Environment – (contd.)

States are 2 types:

- Location state: To help in deciding which direction to move
  [Agent_x, Agent_y, Target_x, Target_y]

- Action State: To help in deciding which action to take
  [Reached Target : 0 or 1,
  Self-Capable : 0 or 1,
  Help on the way : 0 or 1,
  Fire is put off : 0 or 1]

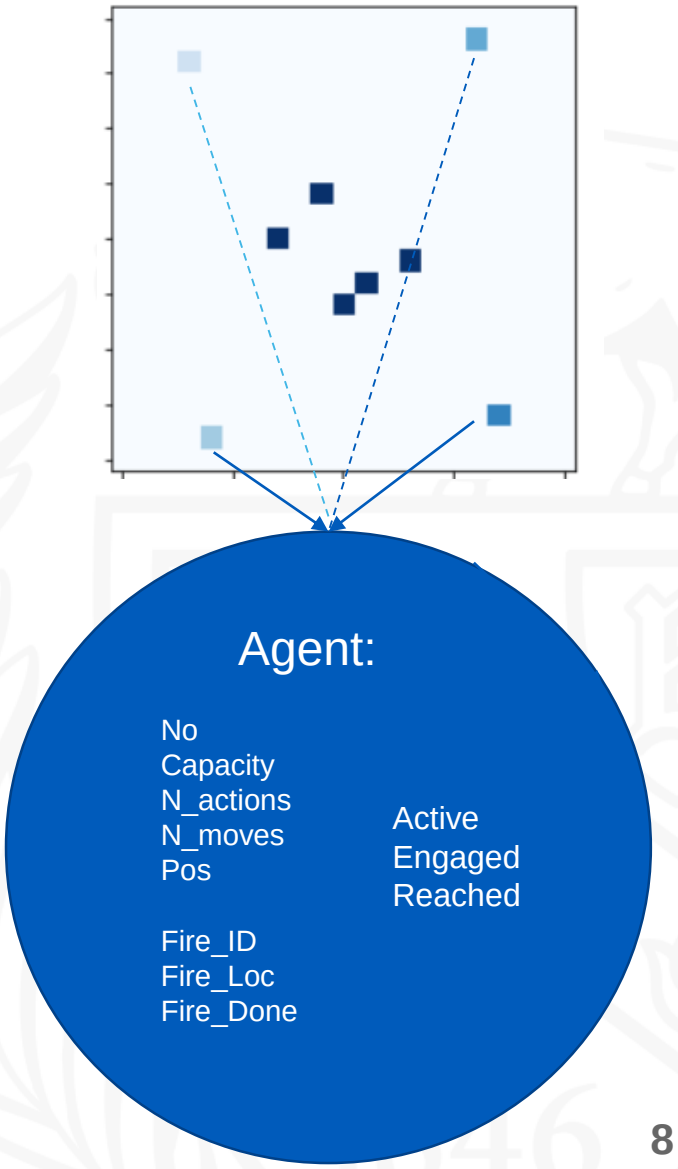
Earlier considerations:
Single state vector : [Agent_x, Agent_y, Target_x, Target_y, Heat_At_Target]

Agent Definition

- Maximum of 4 Agents can be initiated.
- Each agent will be initiated at any one corner
- Each agent knows its status: Active, Engaged, Reached
- Each agent knows its target: Fire_ID, Loc, if fire is off

Action Space

- The degree of movements is 4:
  - Up: 0, Down: 1, Left: 2, Right: 3
- The degree of Actions possible to take is 4:
  - Action 0: Move (up / down / left / right)
  - Action 4: Douse Fire
  - Action 5: Call For Help
  - Action 6: Dis-engage
- Calling for help creates a new message object and stores it in the environment.

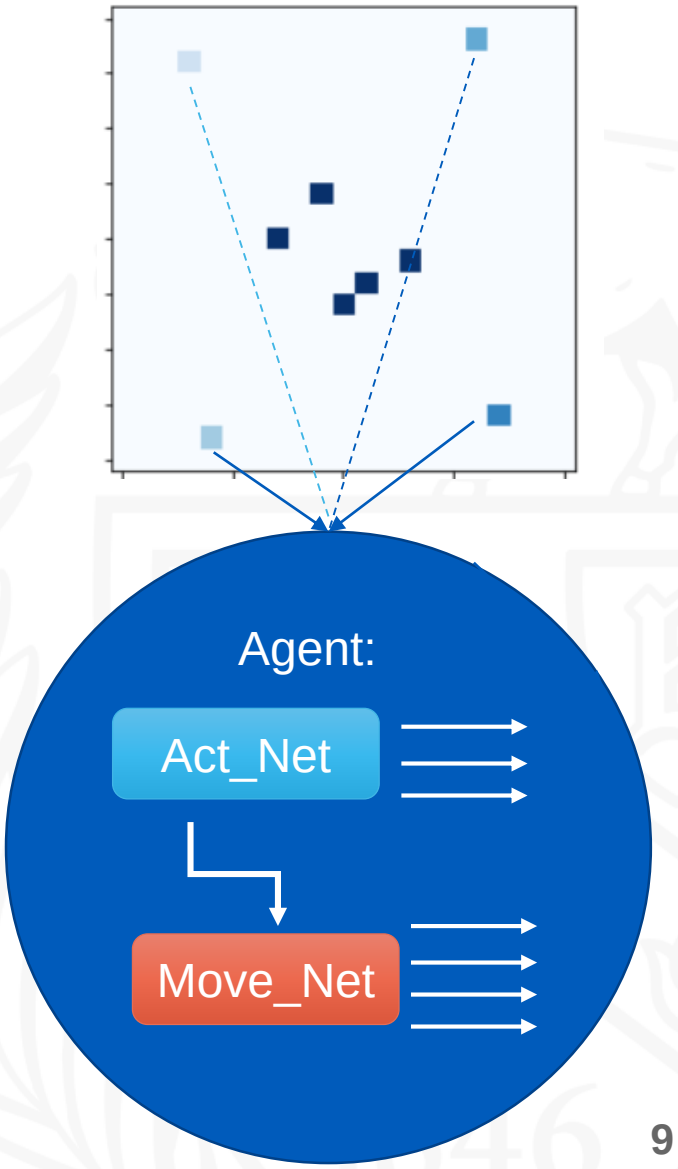


Agent Action Policy:

- Is defined by a DQN named Act_Net
- Takes the Action state as input vector
- Softmax output of 4 different actions
- Training uses Actions Buffer of previous states
- Target Act network is used as in DQN

Action Movement Policy:

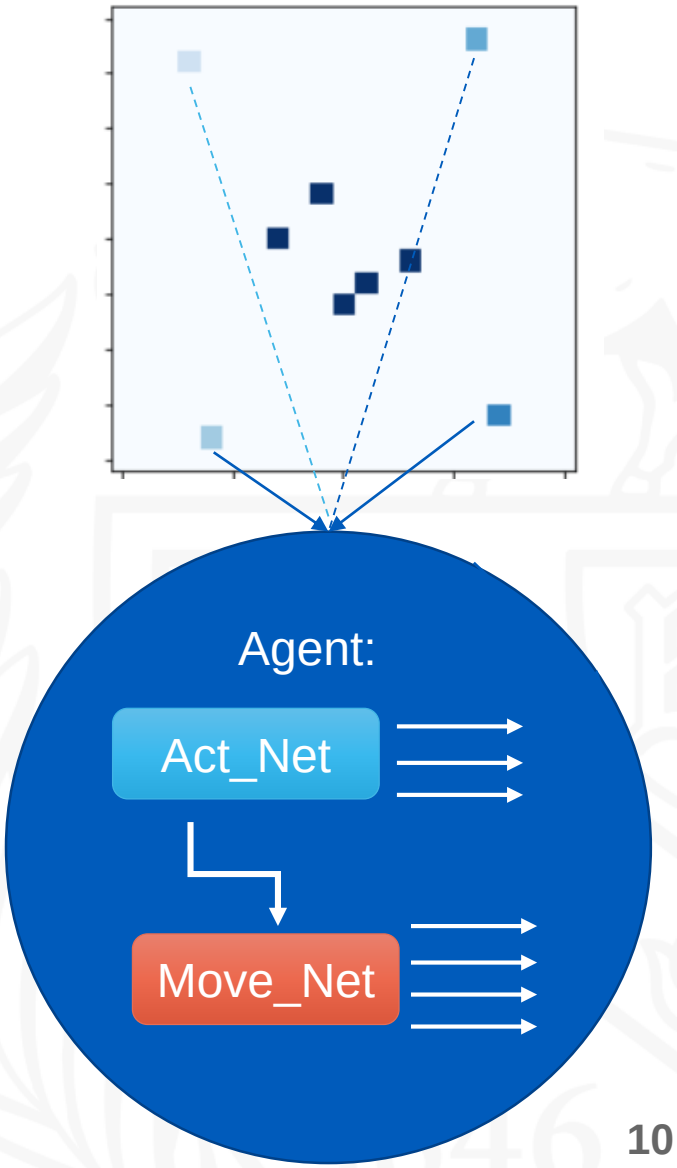
- Is defined by a DQN named Move_Net
- Takes the position state as input vector (coordinates)
- Softmax output of 4 different actions
- Training uses Move Buffer of previous states
- Target Move network is used as in DQN



Vectorized Step:

- Environment spawns few agents and remaining are inactive.
- Each active agent gets Positional State and Action State
- Each action is recorded in a Action vector.
- An individual reward and a common reward are calculated.
- Next state is generated for the agents.
- If an agent calls for help, all agents (including inactive) are queried.

Messages and Fires in the system are checked once again after all actions are implemented by the environment.



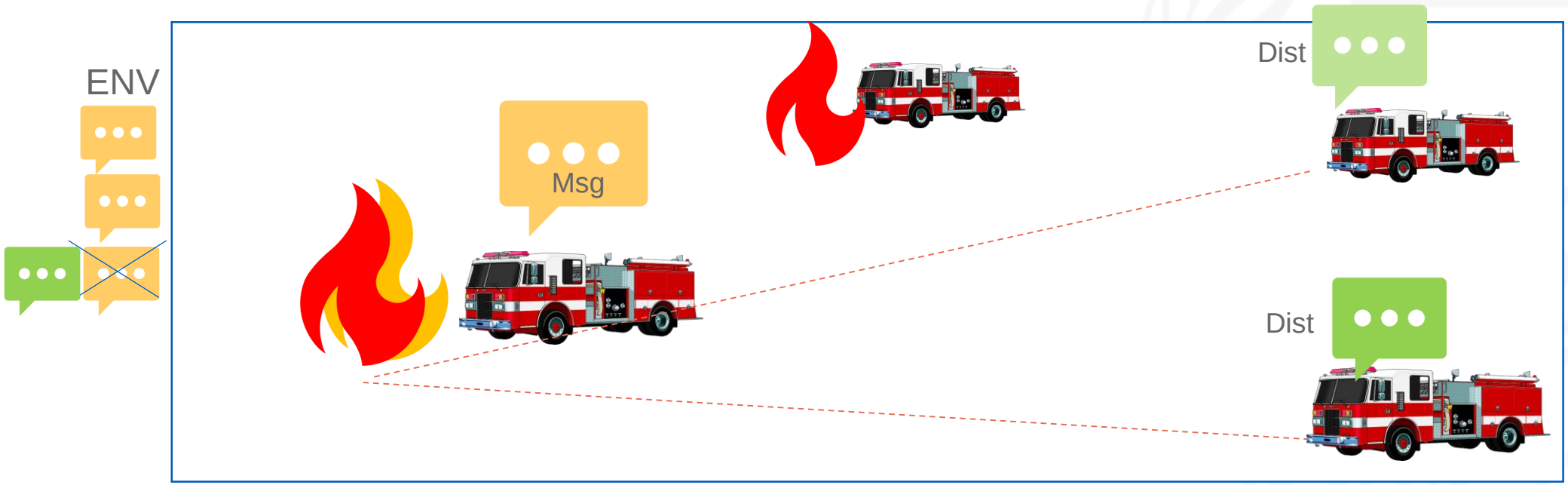
The Message Object – Key for Communication

If an agent detects it is not capable to stop a fire on its own: Action 5 “Call for help”

- Creates a message object with Fire_ID, Fire_Location and if Got_Response

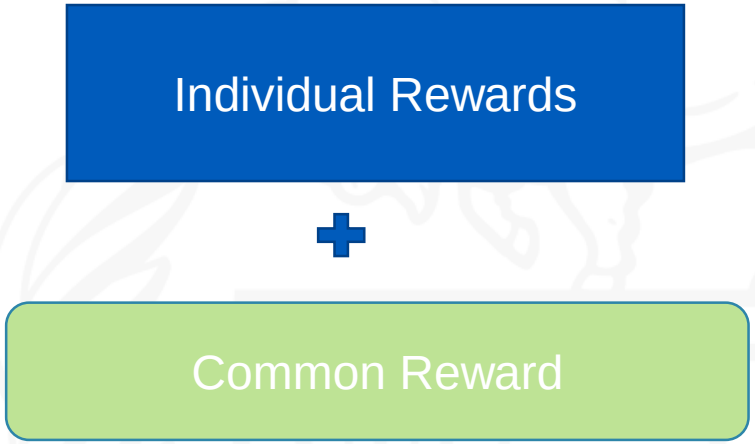
Message

Fire ID
Fire Loc
Got Response



Rewards:

- Moving Closer towards targets: +0.5
- Moving away or staying in same position: -1.0
- Reaching a Target: +2.0
- Not using water jet at target: -2.0
- Using water jet at target: +1
- Calling for unnecessary help: -2.0
- Calling help when not capable: +5.0
- Fire is off at a location: +10 to all agents whose target is that fire



Incremental Learning:

Learn to Navigate:
- Fixed target (Q-Learning)
- Randomly initialized target (DQN)
- Stay at a Target
- Moving closer to target → +ve Reward

Learn to choose to move or douse fire:
- Action 0: Move (if position is not same as target position)
- Action 1: Douse fire (if position is same as target)

Learn to choose to move or different other actions:
- Action 0: Move (if position is not same as target position)
- Action 4: Douse fire (if position is same as target)
- Action 5: Call for help (if heat is > 150) or low capacity
- Action 6: Go Inactive (if fire is put out)

Relaxed Constraints:

- Infinite Capacity Mode: Agent has infinite water capacity:
  - With larger than 5 fires, stochasticity is more. All agents can go out of capacity. Difficult problem to model the environment
- Heuristic Movement:
  - Small grid cannot accommodate many fires. Agents need to be trained again and again for each varying grid size to navigate. For large sized grids, easy to move heuristically and focus on multi-agent tasks.
- Max Steps:
  - Hard to compute number of max steps per each episode. Episode continues until all fires are put out. Max steps based termination is ignored.

Observations:

- Tabular methods are not very good for randomly initiated targets. Can get much faster learning from Neural Net policy
- DQN based navigation is limited by gird size:
  - An agent that learned to move to random target in 5X5 grid finds it difficult to immediately navigate to more distant target if initiated in a larger (say 10 x 10) grid.
  - Moving forward and backward – Needs to see number of experiences to navigate from four corners.
- Shallow networks do perform well:
  - Equations easy to formulate do not need complex neural net architectures. Just enough hidden nodes will do the learning.

Observations (Contd.)

- Proper State Vector matters a lot:
  - Initial choice of state vector was complex involving agent coordinates, target coordinates, heat, capacity and other variables.
  - Even after normalization of features, agent had difficulty to learn a good action sequence.
  - Binary inputs to networks work much better. Pre-processing raw environment attributes into binary state vector helps for better learning.
  - Difficult to learn abrupt changes:
    - Ex. Navigate for 20 steps, stay at one place for next 15 steps. Decision is based on one variable change.

Observations (Contd.)

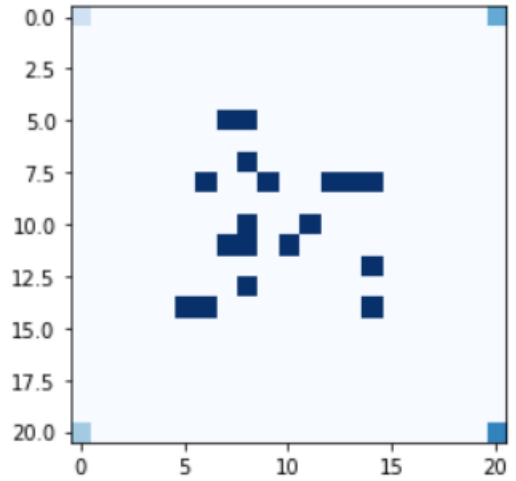
- Impact of Impossible Input Combinations:
  - As agent explores the environment, due to the environment design, some state vectors are never occurring
  - DQN with softmax output layer is essentially a classifier.
  - Gets only partial set of permutations of features if some inputs are not possible to happen.
  - Softmax classifier was not able to learn (accuracy stuck < 70%) on manual testing.
- Solution: Use augmented data in the memory buffer to help improve accuracy.

Reached	Capable	Got Help	Fire Out
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Demo

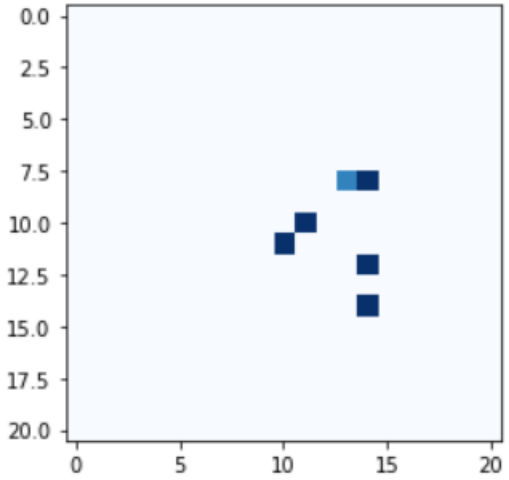
Initial State

Episode: 1



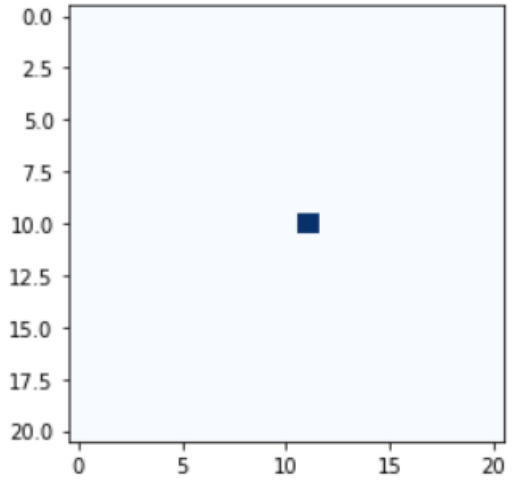

```
Act_State 1 : [0, 1, 0, 0] Action:  0 1002
Act_State 2 : [0, 1, 0, 0] Action:  1 1000
Act_State 3 : [0, 1, 0, 0] Action:  3 1003
Act_State 4 : [0, 1, 0, 0] Action:  3 1001
```

After Some Episodes




```
Act_State 1 : [1, 1, 0, 0] Action:  4 1007
Act_State 2 : [1, 1, 0, 0] Action:  4 1007
Act_State 3 : [1, 1, 0, 0] Action:  4 1007
Act_State 4 : [1, 1, 0, 0] Action:  4 1007
```

Final Episode




```
Act_State 1 : [1, 1, 0, 0] Action:  4 1012
Act_State 2 : [1, 1, 0, 0] Action:  4 1012
Act_State 3 : [1, 1, 0, 0] Action:  4 1012
Act_State 4 : [1, 1, 0, 0] Action:  4 1012
Fire ID: 1012 goes POP
```

Result:

- With incremental learning, Single agent was able to solve a one fire.
- Agent was able to learn to call for help.
- With two or more agents, Agent was able to get help from another agent.
- Multiple agents could simultaneously exchange messages
- Could easily solve fires less than the number of agents.
- Multiple Agents learned to put out as many fires as possible while communicating.
- In theory the agents trained in our environment can putout any number of fires, provided the capacity is sufficient and max steps are allowed.

Scope of Improvement:

- Two DQNs for two kinds of task simplifies the learning complexity, but doubles the memory buffer, training time – A single DQN or other network should be able to solve the task
- Reduce Individual Rewards and increase importance to common reward.
- Switch to a Continuous environment from grid structure
- Include Agents of multiple functionalities (Filler Trucks, Drones for coordination)
- Include obstacles for making environment more complex.

Thank You !