دانشگاه تهران
دانشکده ی مهندسی برق و کامپیوتر
گروه هوش ماشین و رباتیک

Multi-Agent Deep Reinforcement Learning for Fighting Forest Fires

استفاده از یادگیری تقویتی عمیق چند عاملی برای مهار آتش جنگل ها

امیرحسین مصباح
بنفشه کریمیان
عرفان میرزایی

پروژه درس : یادگیری تعاملی
استاد مربوطه : دکتر نیلی

اسفند ماه ۱۳۹۹

Initial Idea

Forest Fires :

- An Important part of natural and Economical damages
- Cost over 1 Billion dollars per year for fighting fires

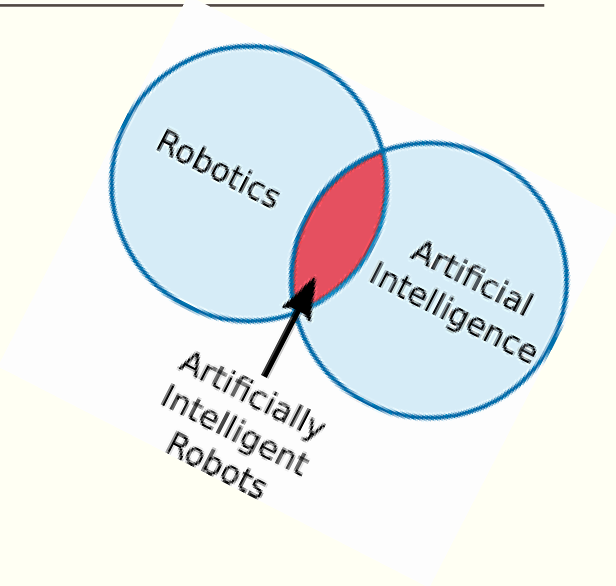
Advantages of solving this problem

- Save lives of firefighters and other humans
- Save Natural Resources and Animal lives

Initial Idea

Solution :

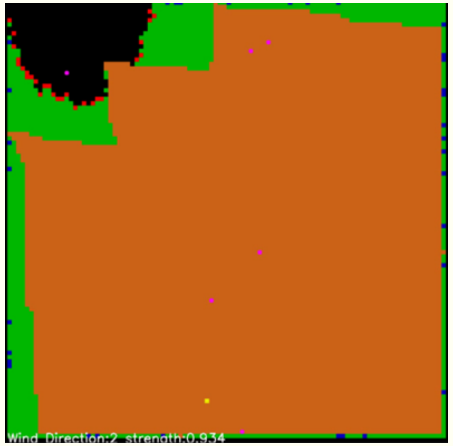
Using intelligent multi-agent robotics



Problem Definition

Main Parts:

Environment



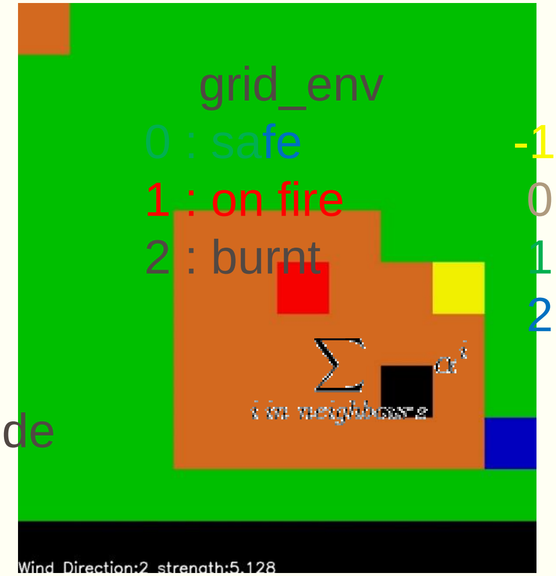
Agents

Learning
Method

Environment

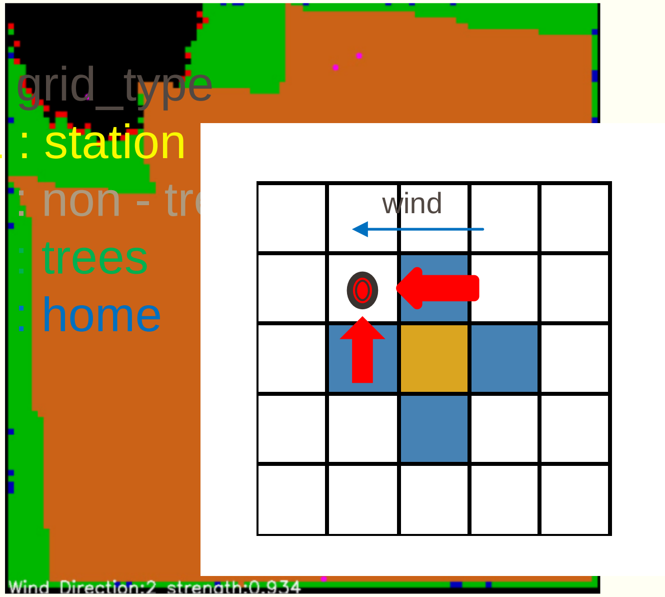
1.Grid with any size
2.gird_env and grid_type
3.Init_fire
4.propagate(wind, table)
5.Terminate

- video capturing from each episode



$\sum_{i \text{ in neighbors}} \alpha^i$
Wind Direction:2 strength:5.128

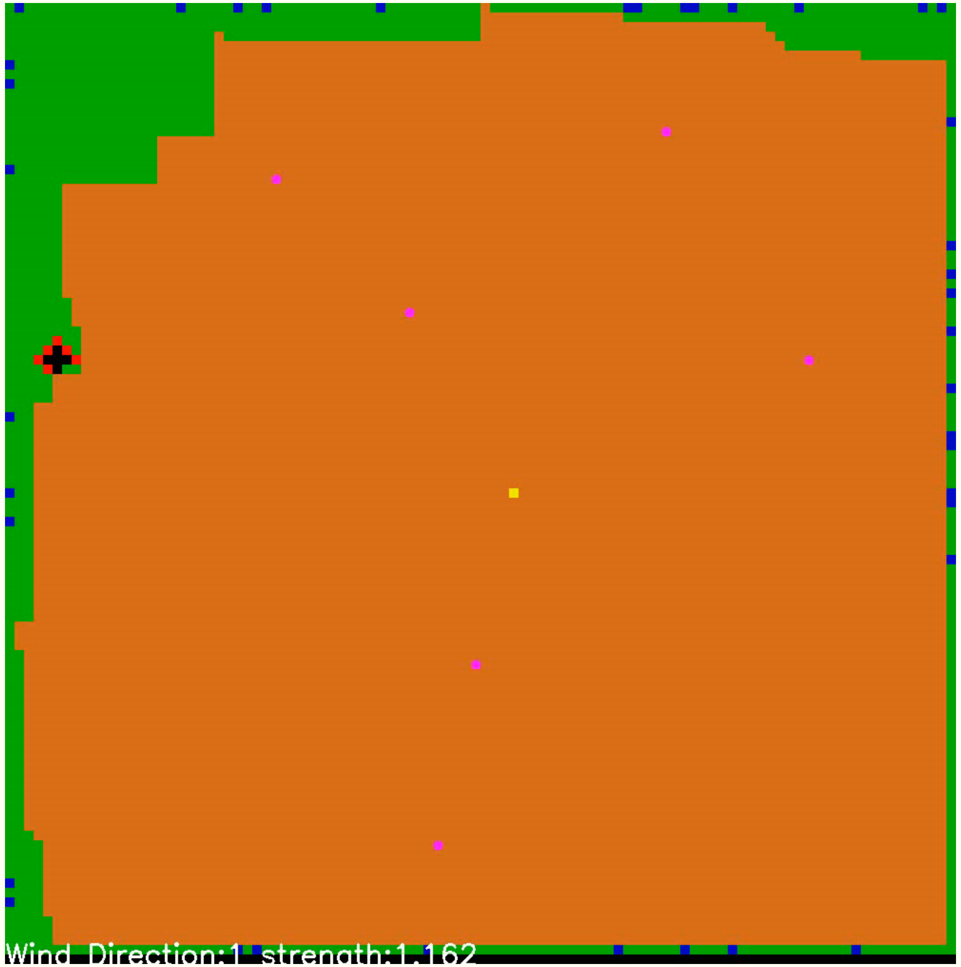
grid_env
0 : safe
1 : on fire
2 : burnt



grid_type
-1 : station
0 : non - tr
1 : trees
2 : home

	Healthy	On-Fire	Burnt
Healthy	1 – P_fire	P_fire	0
On-Fire	0	1 - P_burnt	P_burnt
Burnt	0	0	1

Environment



Wind Direction:1 strength:1.162

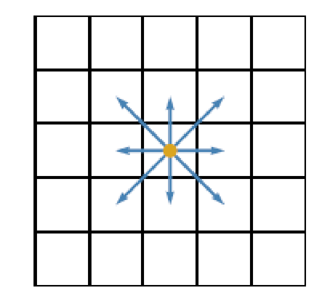
Agents

Simplified model of UAV drones

Actions:

- Fire retardant

- Moving to 8 neighbors



Sensors:

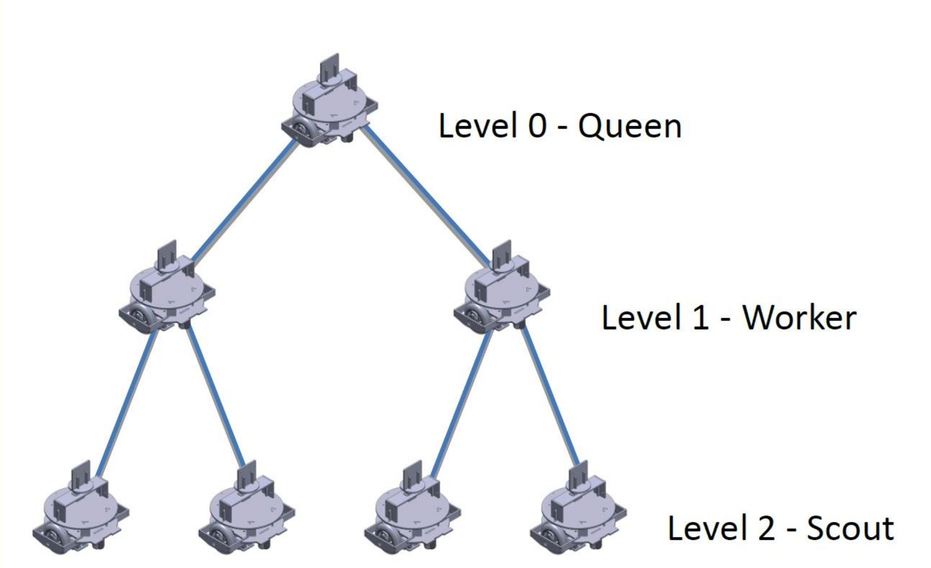
- Camera: 3X3 environment type and 3X3 environment state
- Radio: communication and receive initial mean fire position (updated with camera data)

Agents

Group structure:

Levels:

- Level 0: queen
- Level 1: worker
- Level 2: scout

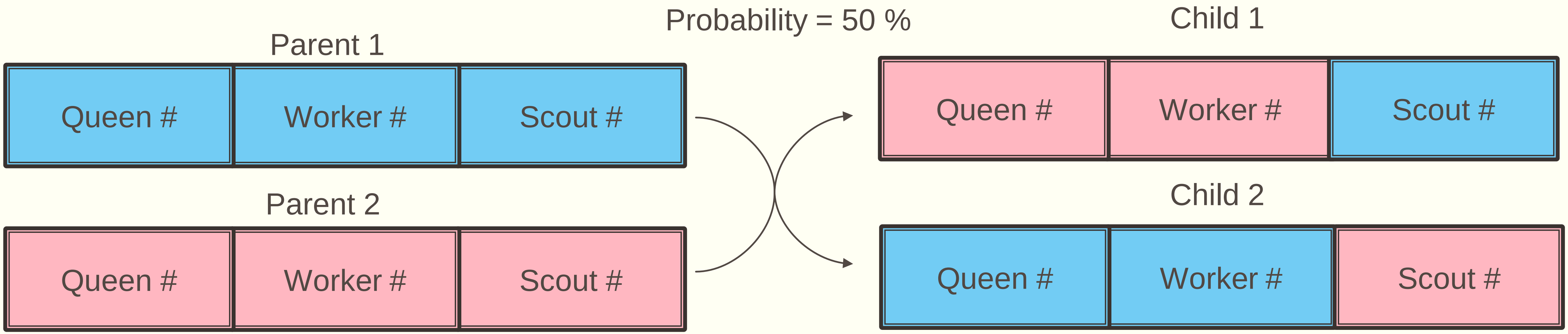


Agents

Finding the best architecture using Genetic algorithm:

- Chromosome:

Queen #	Worker #	Scout #



Probability = 50 %

Parent 1

Queen #	Worker #	Scout #

Parent 2

Queen #	Worker #	Scout #

Child 1

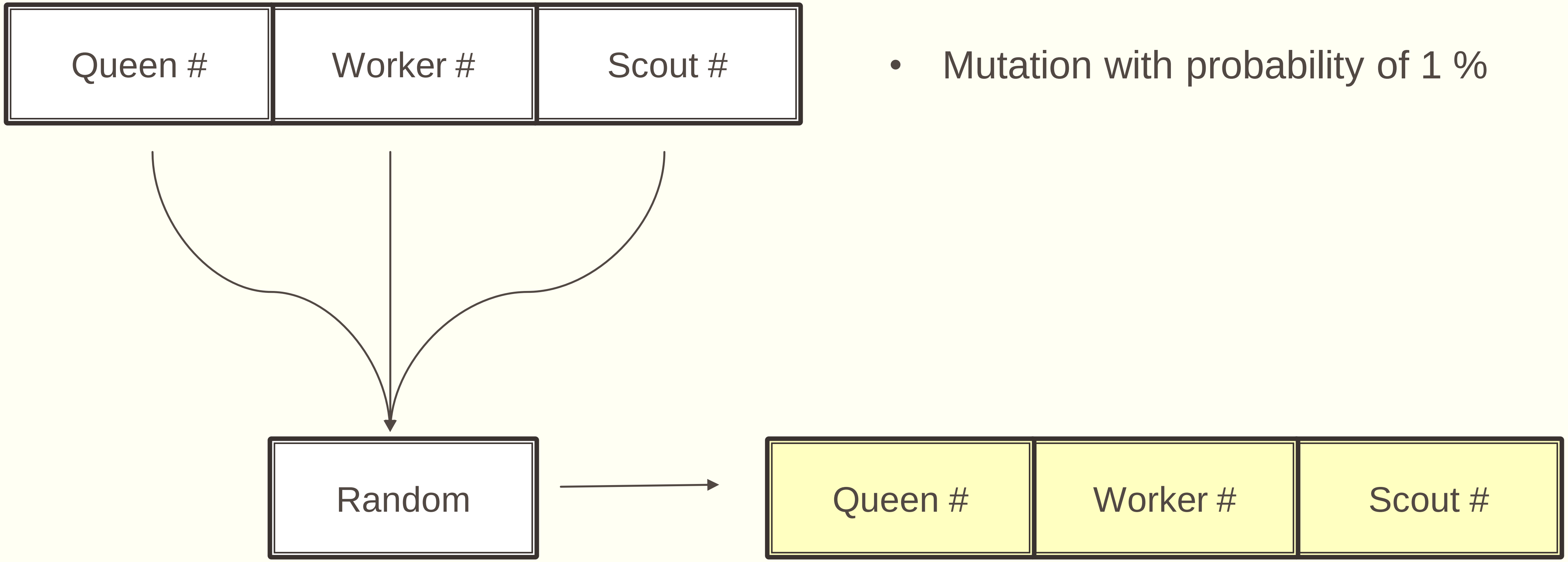
Queen #	Worker #	Scout #

Child 2

Queen #	Worker #	Scout #

Agents

- Mutation with probability of 1 %



Queen # | Worker # | Scout #

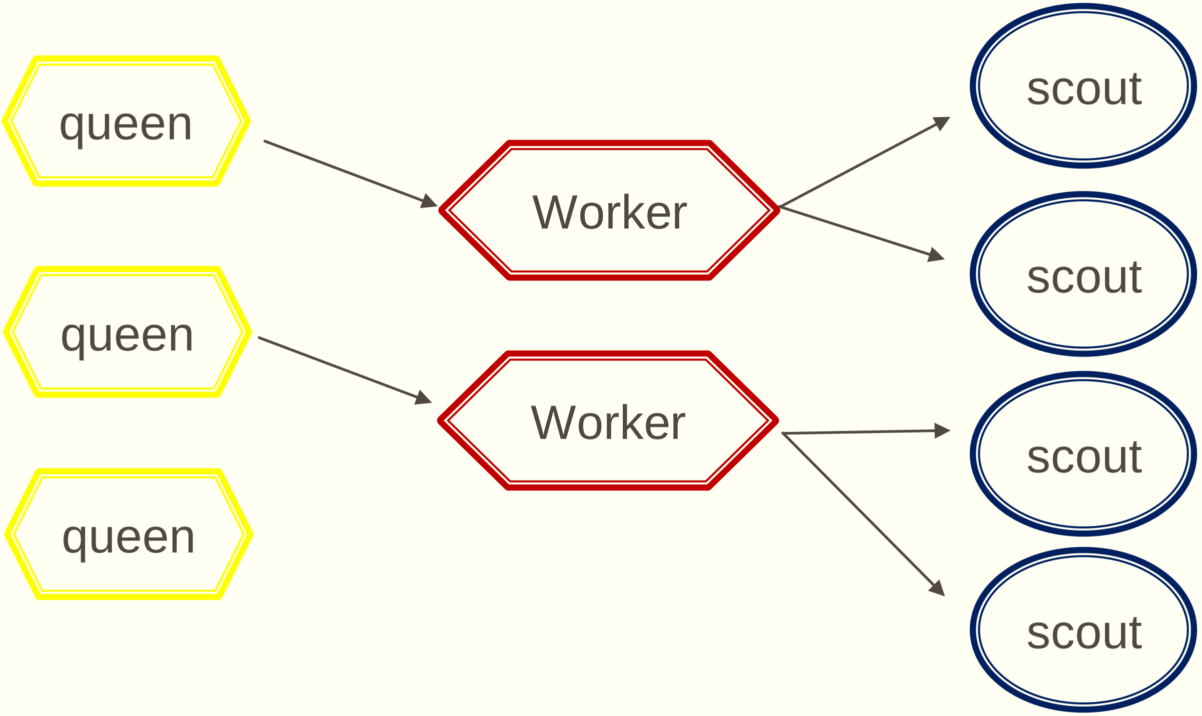
Random

Queen # | Worker # | Scout #

Agents

```
Fitness_function(X):
        make architecture based on X
        initialize Agent
        live n episodes and receive reward   (Pre-trained Network is used)
        return fitness based on received reward
```

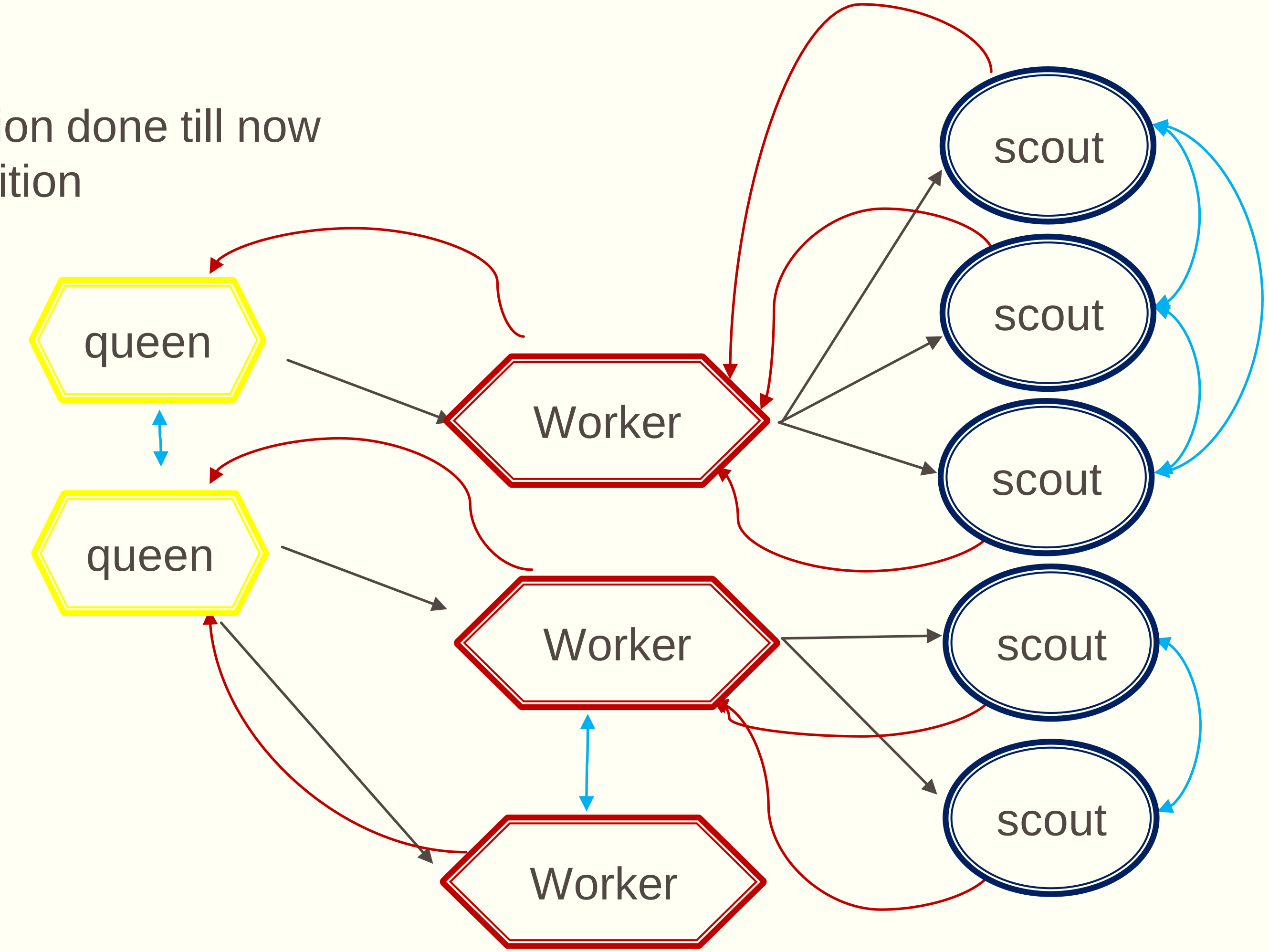
architecture based on chromosome



Agents

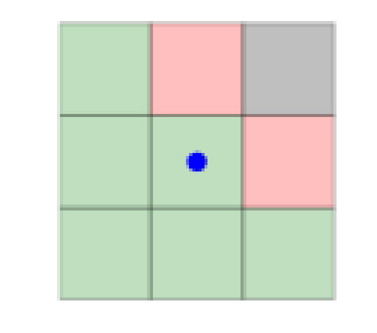
Communication

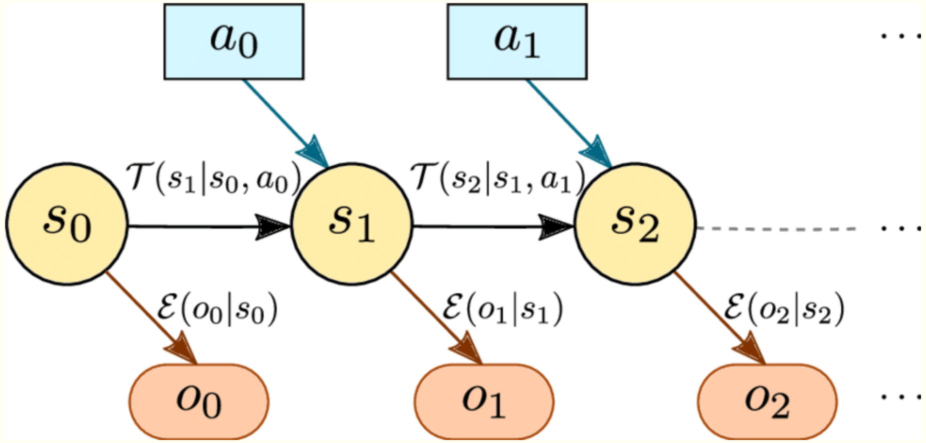
best state action done till now
Mean fire position



Challenges:

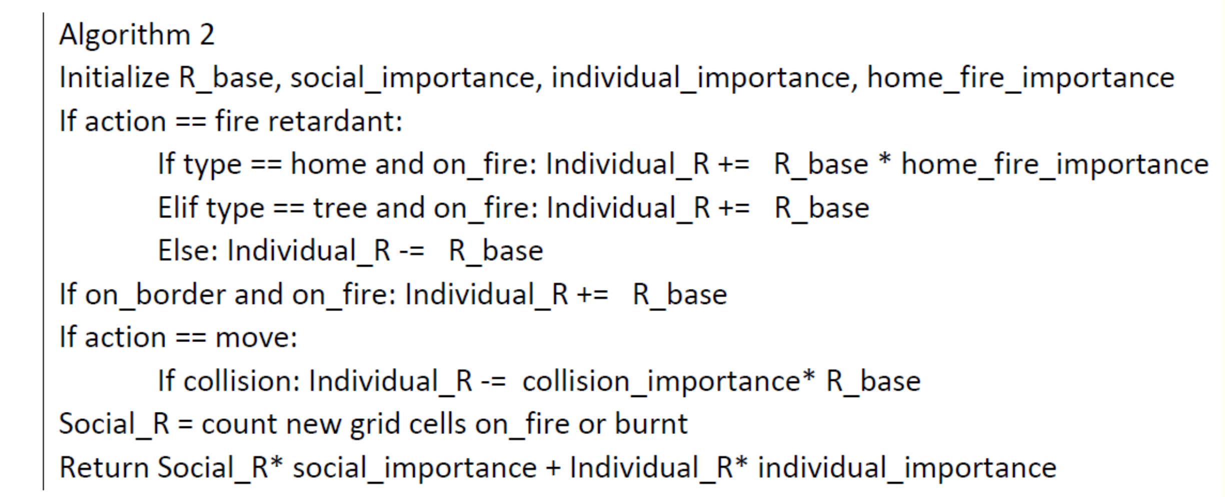
- Partial Observation





- Non-Stationary Environment

- Social Rewarding



```
Algorithm 2
Initialize R_base, social_importance, individual_importance, home_fire_importance
If action == fire retardant:
        If type == home and on_fire: Individual_R +=   R_base * home_fire_importance
        Elif type == tree and on_fire: Individual_R +=   R_base
        Else: Individual_R -=   R_base
If on_border and on_fire: Individual_R +=   R_base
If action == move:
        If collision: Individual_R -=  collision_importance* R_base
Social_R = count new grid cells on_fire or burnt
Return Social_R* social_importance + Individual_R* individual_importance
```

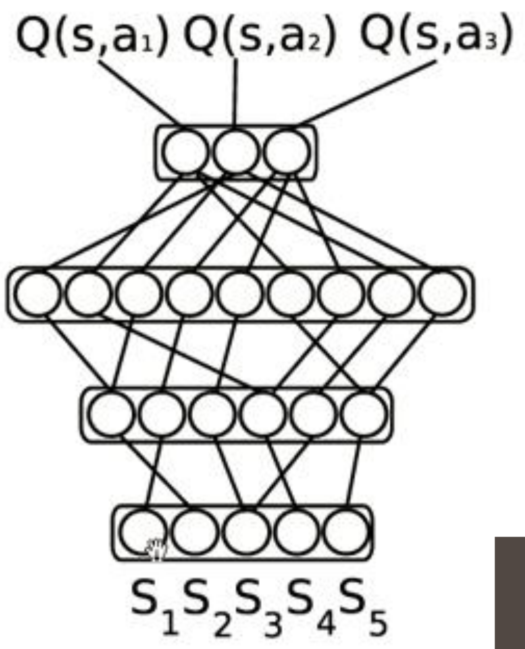
- Large State Action Space

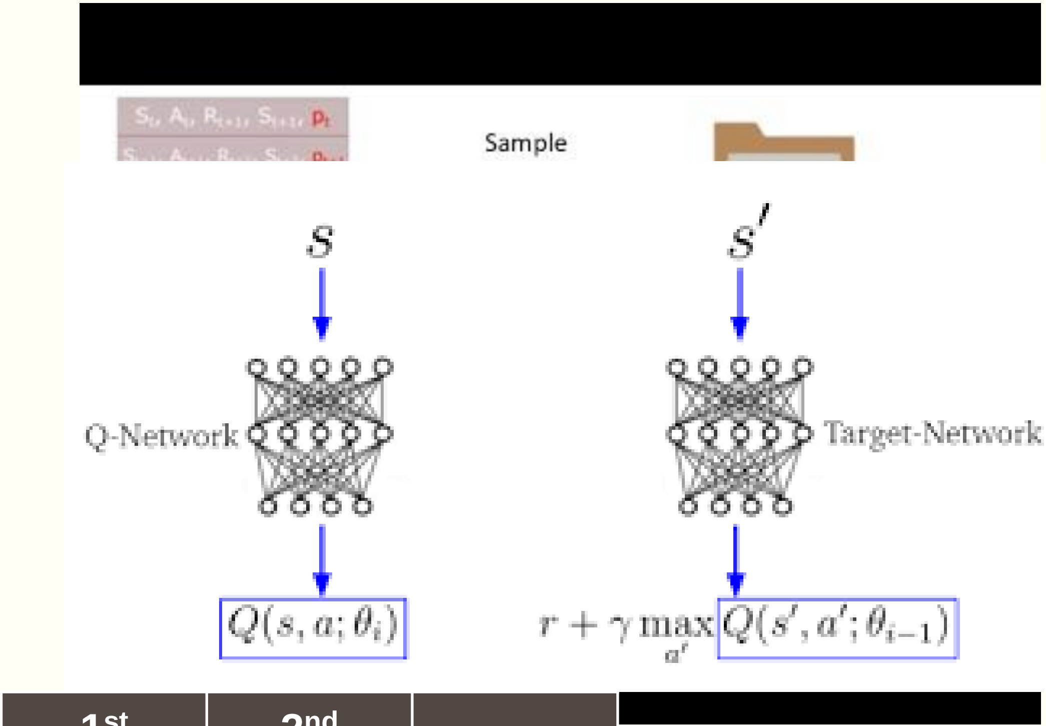
Observation Space ~ $3^{18} * size^4$
Action Space ~ 18

Tabular Methods ❌

Learning Method

Double Deep Q-Network



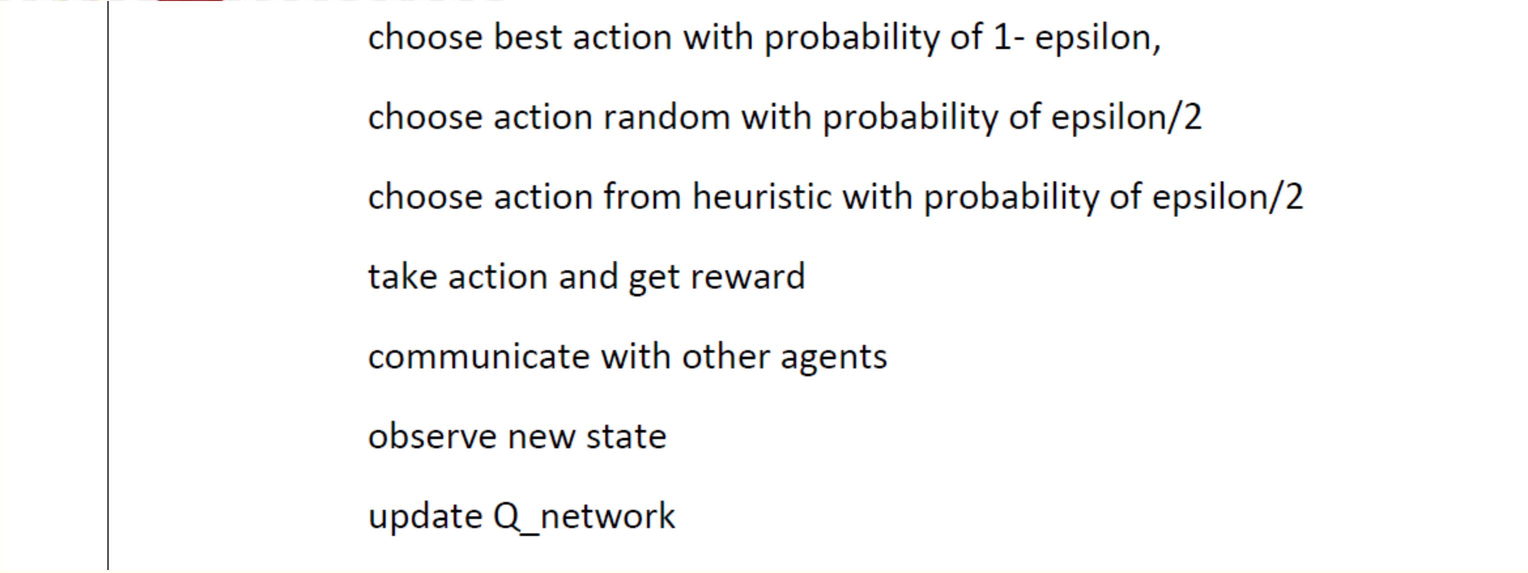


- Networks Architecture

	Input Layer	1st Hidden Layer	2nd Hidden Layer	Output Layer
Q-Network	22	256	256	18
Target-Network	22	128	128	18

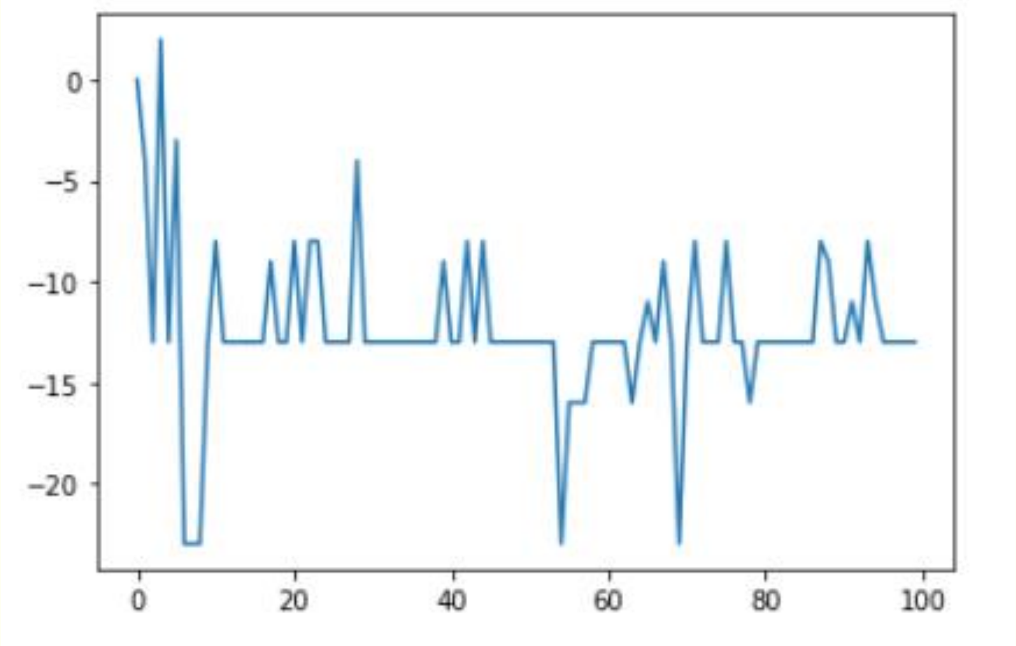
Learning Method

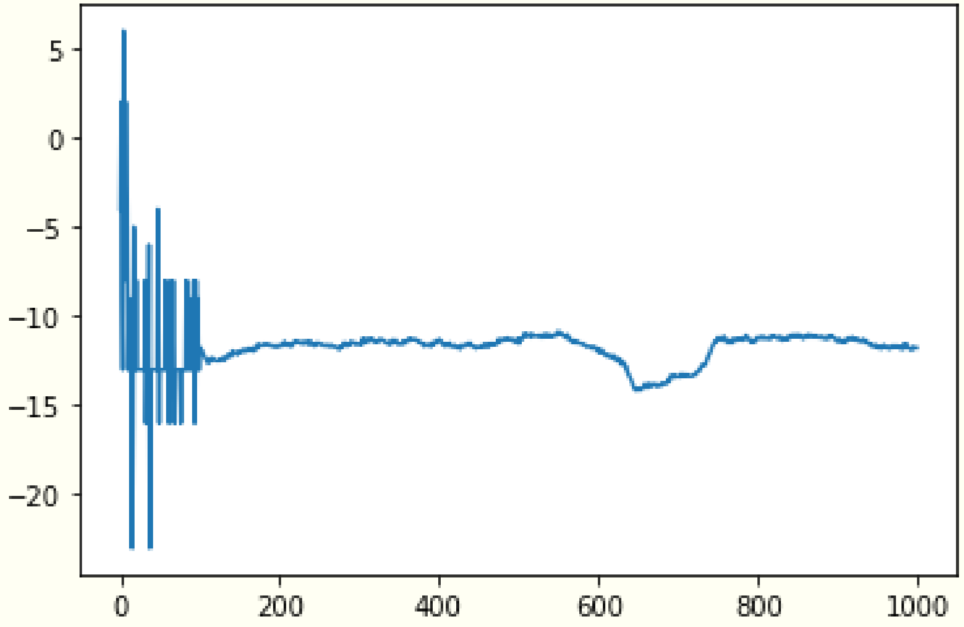
- Epsilon-Greedy
- Heuristic



```
    choose best action with probability of 1- epsilon,
    choose action random with probability of epsilon/2
    choose action from heuristic with probability of epsilon/2
    take action and get reward
    communicate with other agents
    observe new state
    update Q_network
```

Results

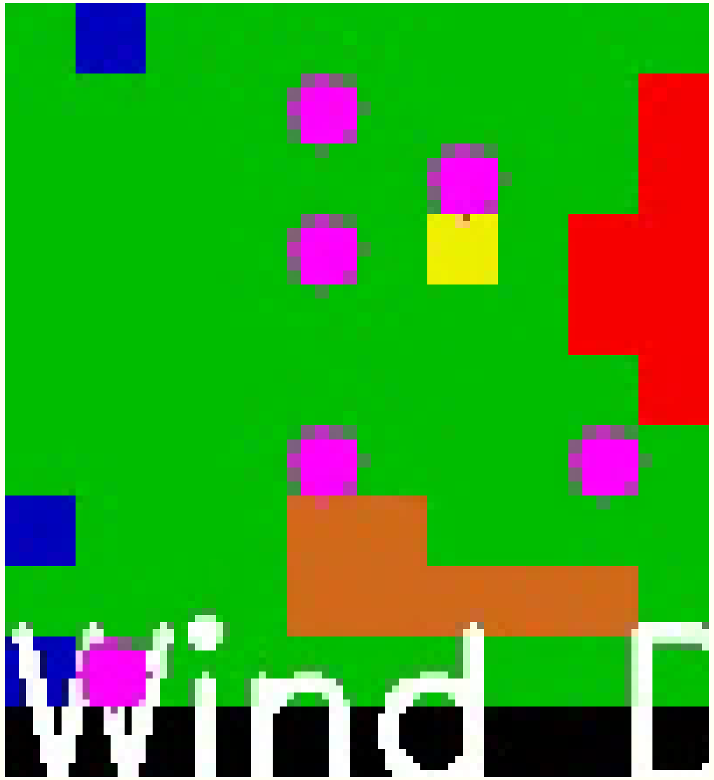


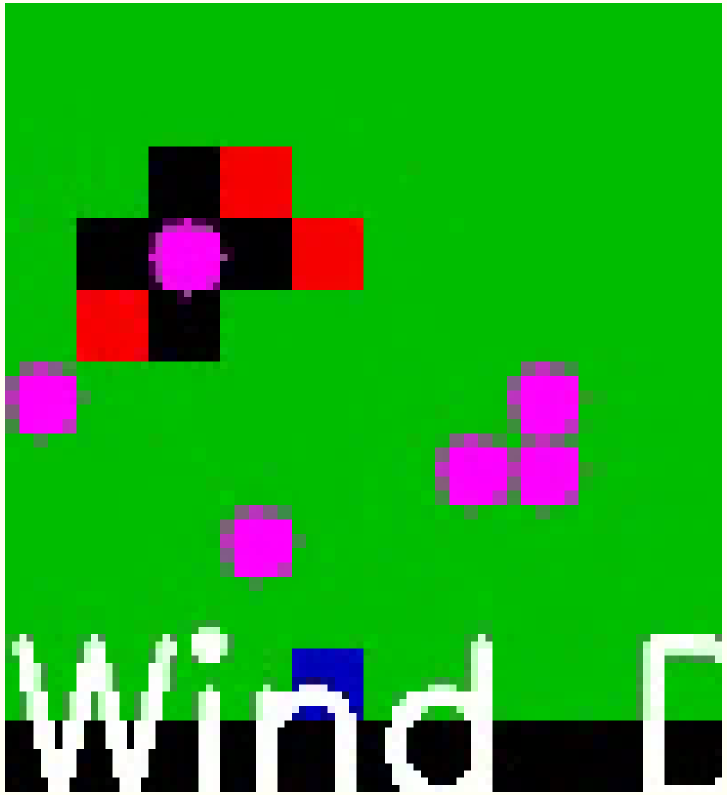


Parameters:
Epsilon = 0.9
Epsilon_dec = 5e-4
Epsilon_min = 0.005
type_plane = [[0,1,2],[0.005,1,0.095]]

Size = 10 * 10
indiviual_reward_importance = 01
social_reward_importance = 0.1
p_change_wind = 0. 1
P_burn = 0.01

Results





Shortage in Computational power :

- Need to train More …
- About 4 hour for 1000 episode on the network

Suggestions for future works

- Limit the capacity of fire retardant materials for agents
- Add Help request to agent actions
- Consider different altitudes for UAVs
- Consider effect of social and individual importance on agent behaviors
- Consider effect of Network Architecture
- Consider different soft policies
- Transfer Learned knowledge for larger environments and it effects on learning speed

Thanks for your attention