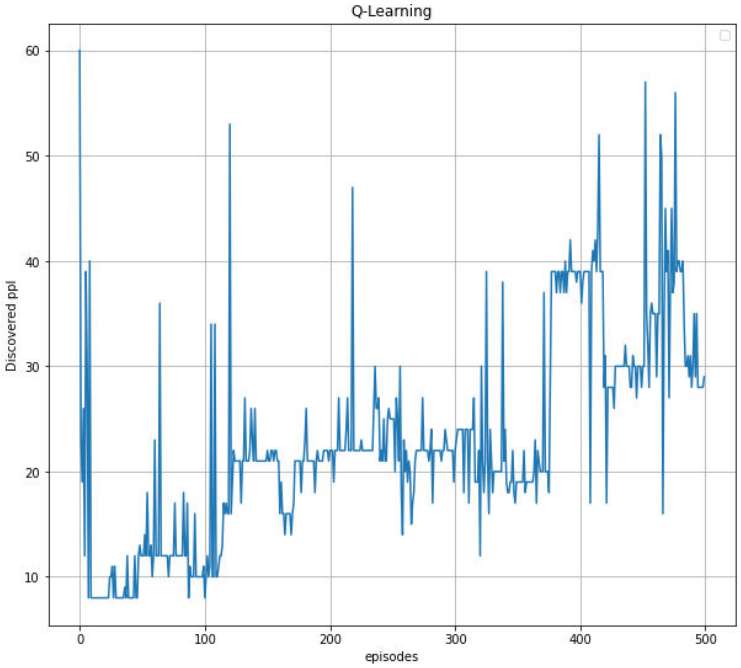
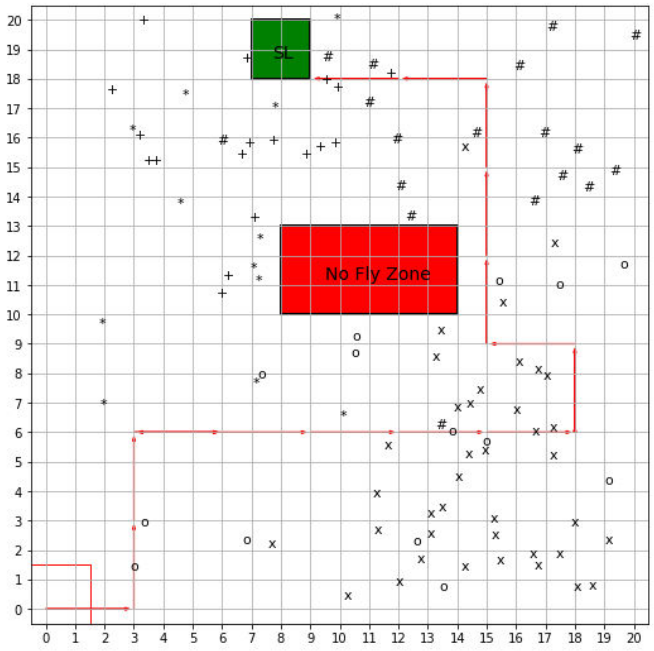
1. Q learning: UAV takes of from (0,0)

a) Number of people found

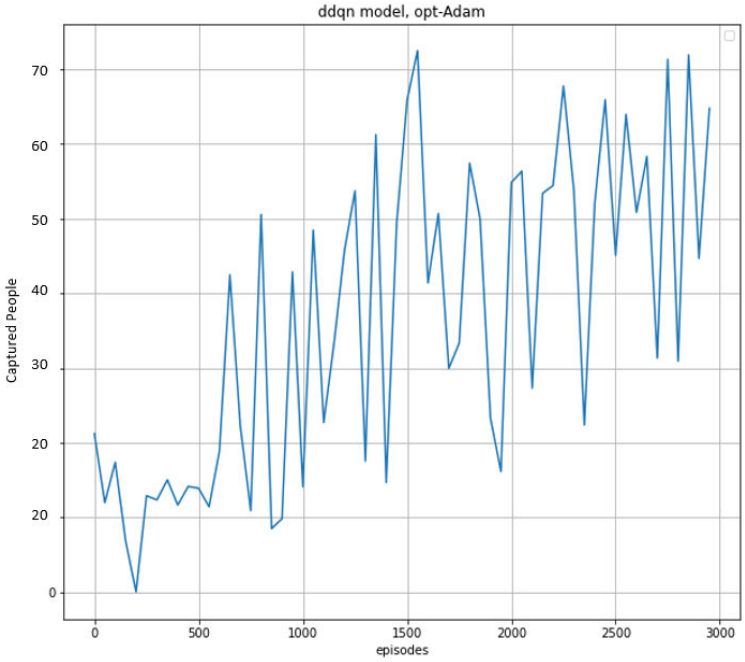


b) Optimal Path

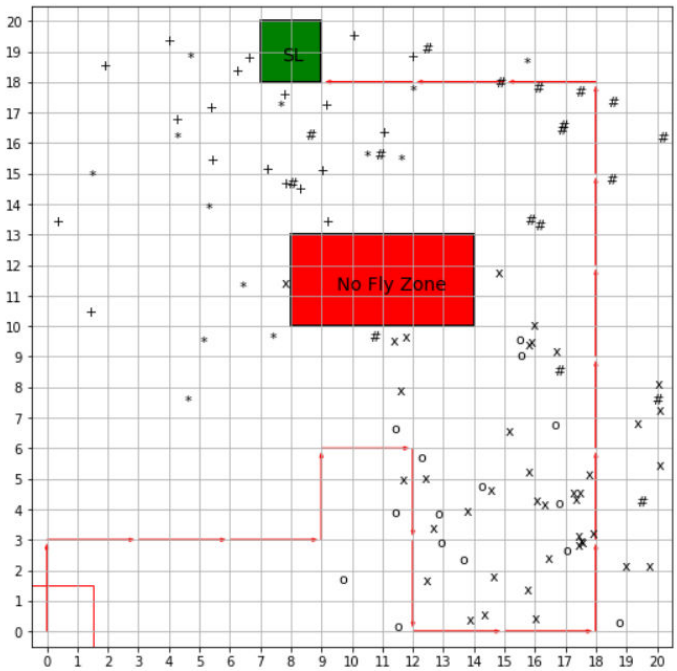


2 DDQN learning: UAV takes of randomly from (x, y) $0 \le x \le 2$ & $0 \le y \le 2$.

a) Number of people found

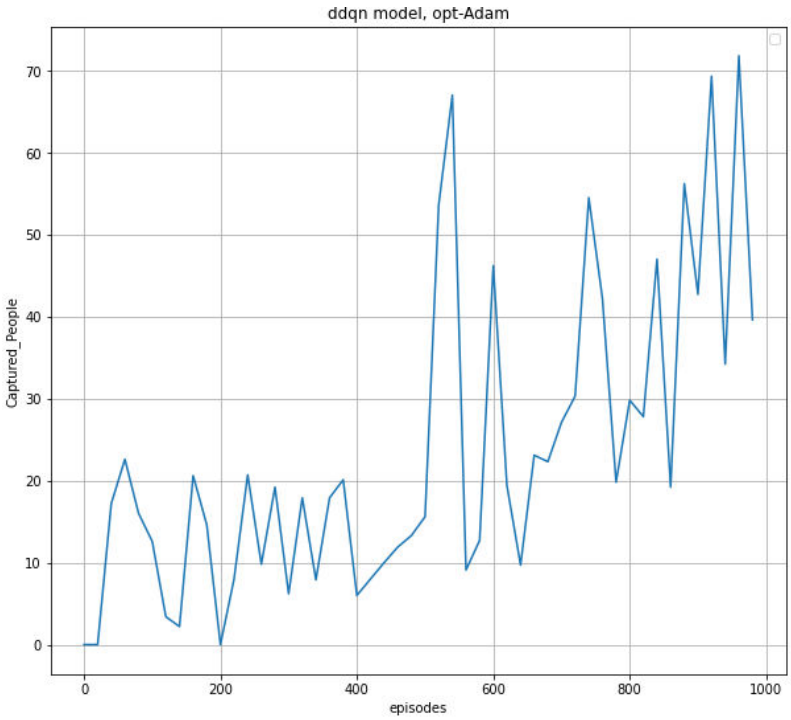


b) Optimal Path

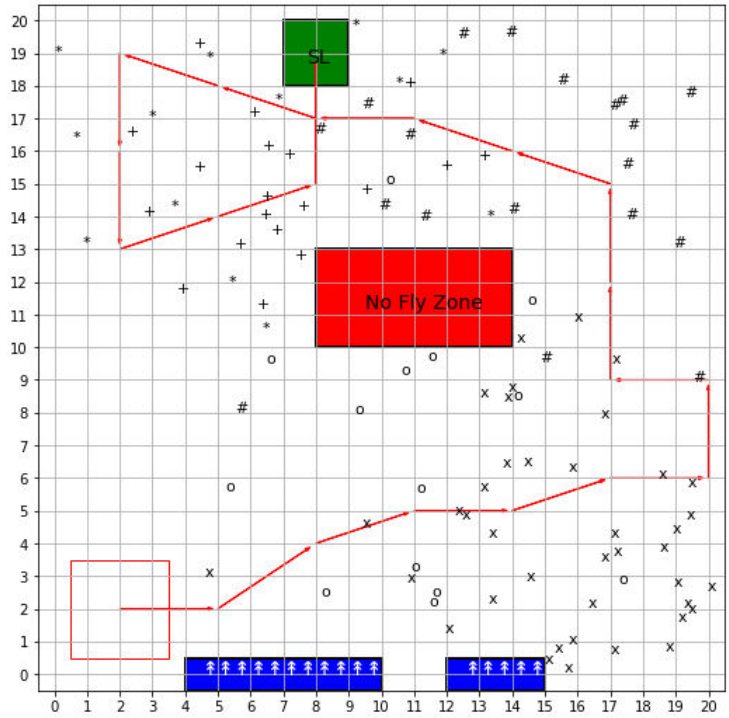


3 Wind Included: DDQN learning: UAV takes of randomly from (x, y) 0 ≤ x ≤ 2 & 0 ≤ y ≤ 2.

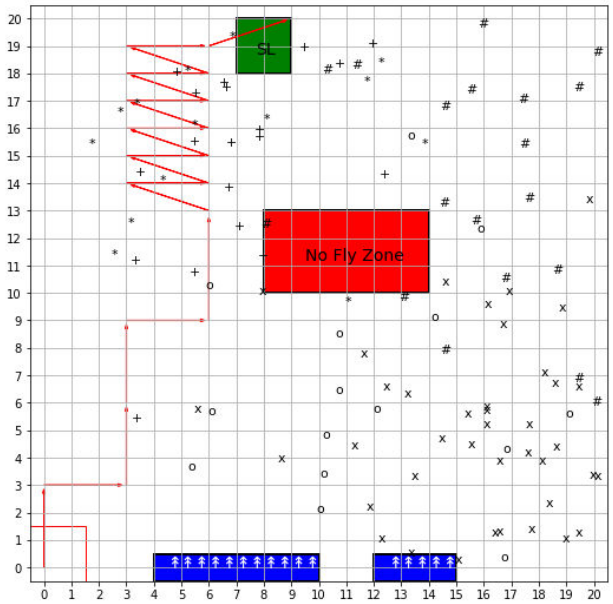
a) Number of people found

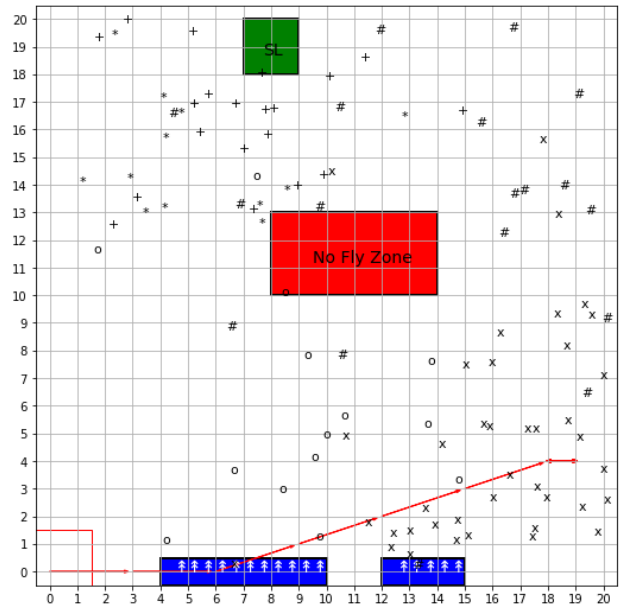


b) Optimal Path



Some other results we got during other run.

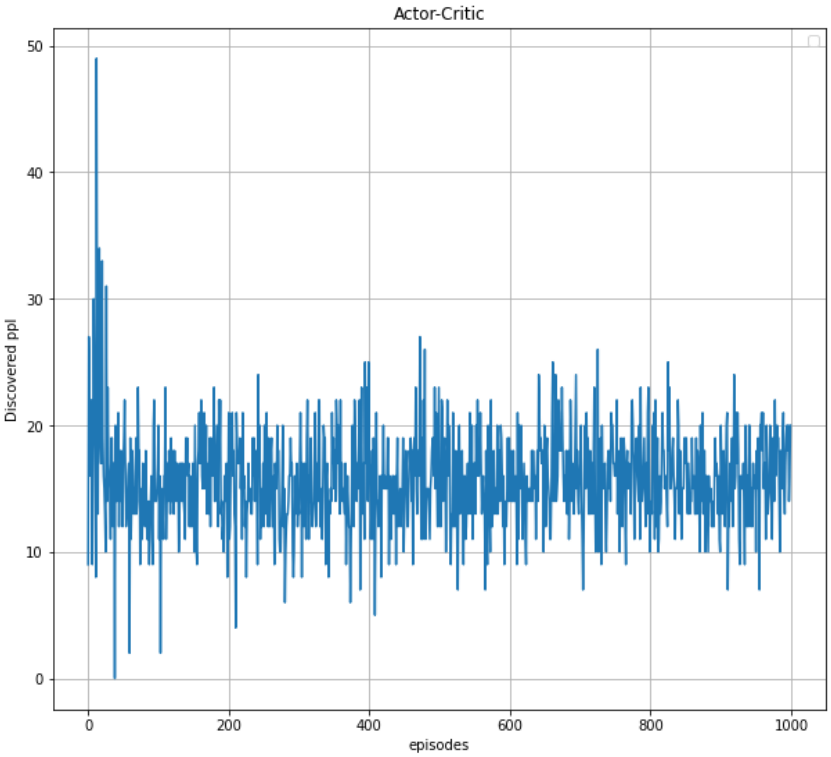




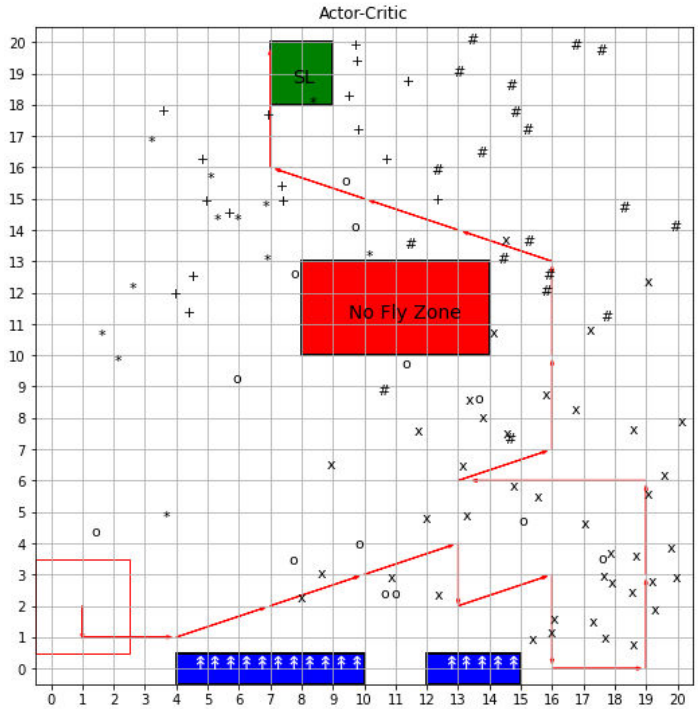
sometimes it gets stuck in sub optimal

4. Actor Critic: user distribution is not fixed: Wind Included: UAV takes of randomly from (x, y) $0 \leq x \leq 2$ & $0 \leq y \leq 2$.

a) Number of people found



b) Optimal Path



Note:

on this project we were able to implement several reinforcement learning technique to maximize the number of users covered by uav while being able to reach the landing site before battery runs out.

This projects needs several hyper-paramters to be tweaked to achiever the best performacne. whether it is the learning rate, number of layers, number of episdoes, epsilon, targeting frequency, optimizer, and several others.

As we can see from the figures above. we were able to see convergence of the RL techniques however there were inconsistancy and hard made to reproduce some optimal results sometimes.

We believe those inconsistency could be achieved by having a well constructed reward and punishment as well as right hyper-parameters