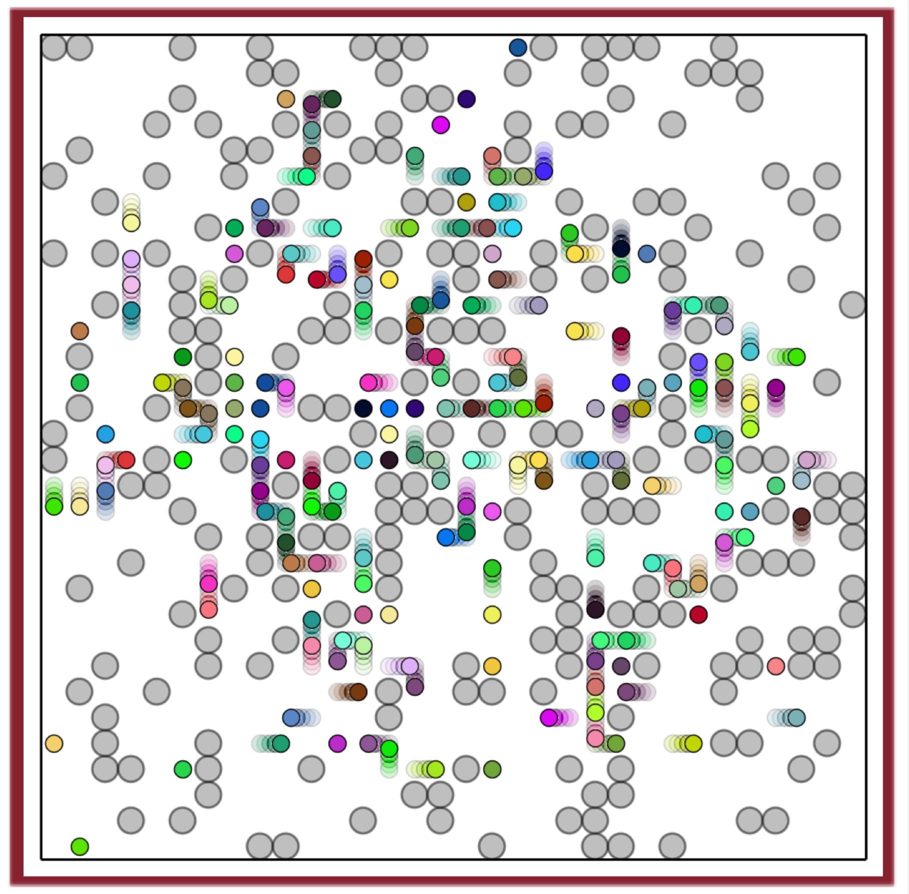
Policy Optimization for Multi-Agent Path Finding

Anthony Flores-Alvarez, anthonyflores630@gmail.com
Manual Arts Senior High School, Class of 2021
USC Viterbi Department of Computer Science, SHINE 2020

Introduction

Multi-Agent Path Finding (MAPF) is the problem of finding paths for multiple agents such that every agent reaches its goal without colliding with one another. In recent years, there has been a growing interest in MAPF in the artificial intelligence (AI) research community because real-world MAPF applications, such as warehouse management and multi-robot teams, are becoming much more prevalent. However, current state-of-the-art MAPF planners are slow. This is because MAPF is an NP-hard problem, meaning you cannot arrive at an optimal solution in polynomial time. To resolve this, Ph.D. student Eric Ewing and I built upon a current state-of-the-art MAPF framework [1] and reinforcement learning (RL) algorithms to design a faster MAPF planner using RL and a DNN.

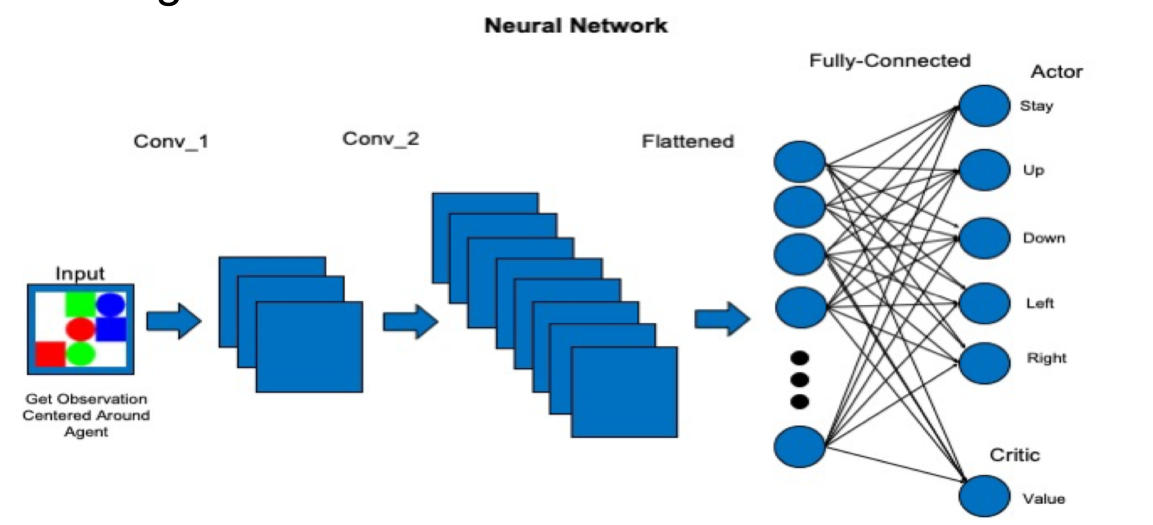

[2]

MAPF Environment

Methods

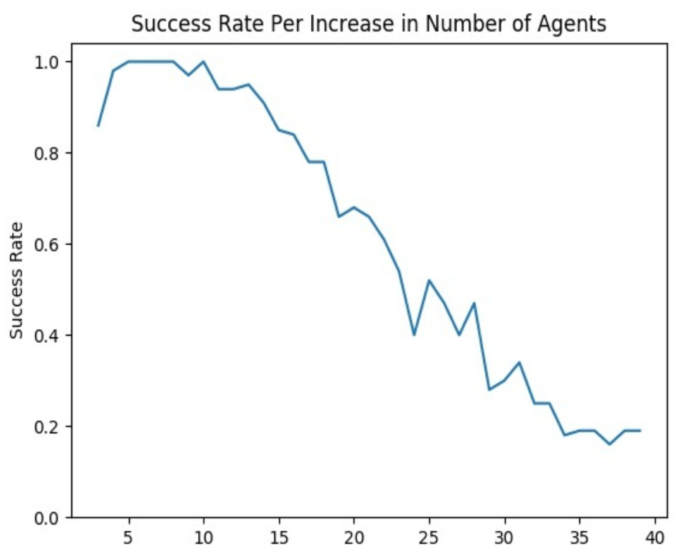
To begin developing our MAPF planner, we began with identifying the observations that our deep neural network (DNN) would take as inputs. The observations in our environment were the locations of the obstacles, agents, and goals present in the limited field of view (FOV) of our agent's observation space, as well as a vector pointing to a specific agent's goal and the vector's magnitude, as a natural way to let agents learn to select their general direction of travel. After this, we began building our DNN's architecture.

Advantage Actor Critic (A2C) allows us to maintain the best of policy and value-based RL algorithms, as well as capture how better an action is in comparison to others, at a given state with the addition of the advantage function.



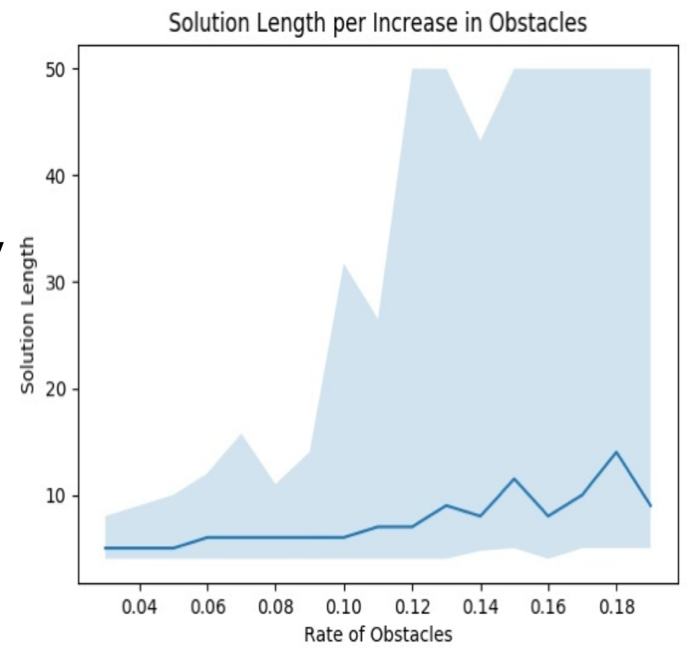
We trained the model using the A2C algorithm. Once it was trained, we ran the model on a few experiments where we either varied the size of our environment, obstacle rate, or varied the number of agents. We did this to calculate our framework's success rate when we varied the number of agents, as well as the solution length when we varied the obstacle rate or environment size .

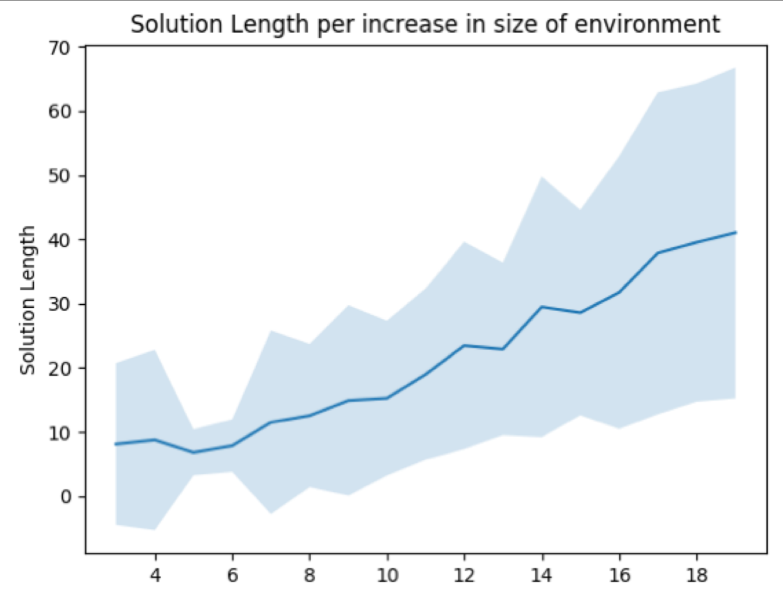
Results



This graph displays success rate per increase of the number of agents in the environment. The key takeaway is that as you increase the number of agents the environment becomes progressively harder to solve, as shown with a downward trend. This is expected as MAPF is an NP-hard problem.

This graph displays solution length per increase in the obstacle rate. The key takeaway is that solution length increases because more obstacles make solving the environment more difficult.





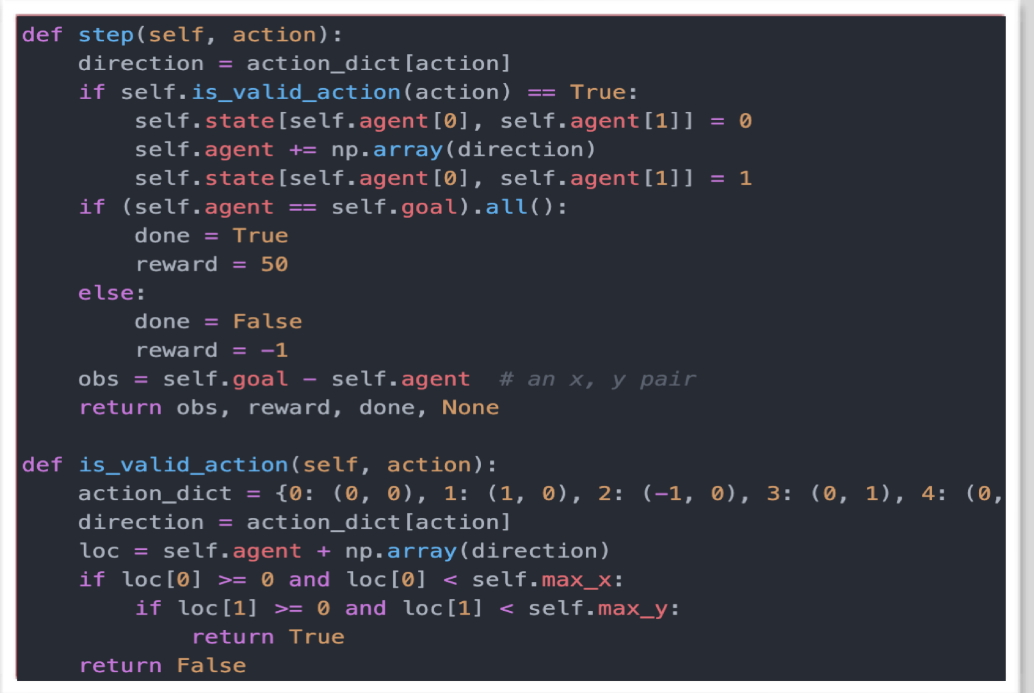
SCAN ME

MAPF Environment Solution Example

This graph displays the solution length per increase in the size of the environment. The key takeaway is that the solution length increases because there are more steps for agents to take until they reach their goal which is caused by the increase in environment size.

Skills Learned

- Neural Network Development
  - Build, train, explore, debug neural networks
  - Designed neural network to classify MNIST handwritten digits data set
- Software Experience
  - Used Numpy, Pytorch, OpenAI Gym, Atom
- Reinforcement Learning Experience
  - Used A2C for MAPF policy optimization and Q-Learning for multi-armed bandit solution
- Advanced Python Programming Literacy
  - Programmed neural networks and RL algorithms using prior Python experience, Python libraries, and object-oriented programming (OOP)
- MAPF Experience
  - Optimize policies for MAPF by using different RL algorithms and DNN structures
- OpenAI Gym - RL Algorithm Testing Experience
  - Used Q-Learning and A2C algorithms for Cart-pool solutions
  - Used Imitation Learning (IL) Heuristic for LunarLanderV2 solution
- Artificial Intelligence Mathematics
  - Studied Markov Decision Processes for Multi-Armed Bandit Problem Solution
  - Studied probability distributions and Bayesian statistics to comprehend and use the mathematical definitions of the RL algorithms used



```python
def step(self, action):
    direction = action_dict[action]
    if self.is_valid_action(action) == True:
        self.state[self.agent[0], self.agent[1]] = 0
        self.agent += np.array(direction)
        self.state[self.agent[0], self.agent[1]] = 1
    if (self.agent == self.goal).all():
        done = True
        reward = 50
    else:
        done = False
        reward = -1
    obs = self.goal - self.agent  # an x, y pair
    return obs, reward, done, None

def is_valid_action(self, action):
    action_dict = {0: (0, 0), 1: (1, 0), 2: (-1, 0), 3: (0, 1), 4: (0,
    direction = action_dict[action]
    loc = self.agent + np.array(direction)
    if loc[0] >= 0 and loc[0] < self.max_x:
        if loc[1] >= 0 and loc[1] < self.max_y:
            return True
    return False
```

Building MAPF Environment|PC: Anthony Flores

Next Steps for Me & References

I will build upon my SHINE research by joining a robotics and/or artificial intelligence lab during my undergraduate years at the college I attend next fall. There, I will research more MAPF frameworks and planners, as well as deep RL and non-ML algorithms which will aid me in one day designing a novel MAPF framework.

[1]. Sartoretti, G., Kerr, J., Shi, Y., Wagner, G., Kumar, T. K. S., Koenig, S., & Choset, H. (2019). PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning. *IEEE Robotics and Automation Letters*, https://doi.org/10.1109/lra.2019.2903261

[2]. Felner, Ariel, et al. *AAAI 2017 Tutorial SUP3: Introduction to Multi-Agent Path Finding*, 5 Feb. 2017, www.andrew.cmu.edu/user//gswagner/aaai_tutorial/aaai_2017_tutorial.html.

Acknowledgements

A sincere thank you to Professor Nora Ayanian for accepting me into the ACT Lab, and a special thanks to my Ph.D. mentor Eric Ewing for being an incredible mentor, patiently teaching me about RL despite its complexity, and for inspiring me to pursue CS research. Also, thank you to my lab partner Vrinda Bansal and Center Mentor Ashley Perez for making my SHINE experience truly unforgettable. Last, but certainly not least, thank you to Dr. Mills for reminding me that I did not have to choose between CS or politics and for always being very welcoming and fun!