Work Plan – C3

Petter Eriksson, Daniel Dalbom

February 2019

Background

The following document details the work plan for project C3: Distributed Optimization through Deep Reinforcement Learning. The purpose of the project is to implement within multiple agents a deep reinforcement learning algorithm. The agents should cooperate to learn how to interact and accomplish some given tasks optimally. More specifically, we are to focus on a simple warehouse, where multiple robots need to transport cargo from one point to another, in the shortest time possible, while avoiding collisions with each other and other obstacles.

Goals

The overall goal of the project is to simulate multiple agents transporting items from one point of a warehouse to another while avoiding collisions, using a deep reinforcement learning algorithm for finding the optimal paths. The simulation will be performed using the Python 3 programming language. In order to succeed with this, the following subgoals need to be achieved:

Description of subgoals

- *Implement a suitable environment for one agent.* Model and implement the warehouse, complete with obstacles, walls and items to pick up.
- *Implement a single agent using a reinforcement learning algorithm.* The agent should gradually learn to avoid obstacles while finding the optimal path, i.e the path of fewest steps, between two points, A and B. To verify that this is working, simulate the system and plot how the number of steps needed for moving between A and B changes with time. If it decreases and eventually converges towards a minimum, the algorithm should work as intended.
- *Implement two agents using a reinforcement learning algorithm.* The two agents should be able to move around the warehouse between desired points gradually learning to avoid collisions with obstacles and each other.

Again, plot the numbeŕ of steps for each agent as a function of time to see how it changes. Also, record the number of collisions between the agents and examine whether it decreases over time as it should.

- *Implement multiple agents using a reinforcement learning algorithm.* Generalize the algorithm in order to implement an arbitrary number of agents. The system should fulfill the same requirements as for two agents.
- *Implement a neural network.* To deal with more complex systems, such as a large number of agents or a warehouse of large size, neural networks will be used for data processing. A first step is to implement a simple neural network and understand how it works. More research about neural networks is needed before deciding on testing methods.
- *Integrate the neural networks with the reinforcement learning algorithm.* The resulting system should be able to simulate multiple agents moving around a grid of decent size. The system should fulfill the same requirements as for multiple agents.

Organization

The project team consists of:

- Daniel Dalbom, ddalbom@kth.se; Armégatan 32c, Solna, Sverige
- Petter Eriksson, petteer@kth.se; Kungshamra 21, Solna, Sverige

The responsibilities are divided equally between the project members, who all contribute to every subtask.

Process

Project phase	Milestone	Ready date
Planning	Project plan ready	15/2-2019
Implementation	Multiple agents implemented using neural networks	7/4-2019
Report writing	Final draft finished	20/5-2019
Presentation	Presentation held	16/5-2019

Communication

The communication with the supervisors will be done through continuous e-mail correspondence. Weekly meetings will be held with the supervisors. During these, the progress is reported and future problems and steps needed to resolve them are discussed.

Risks

In the following section, the risks associated with each subgoal are detailed. If they are preventable, a proactive measure is included.

- *Implement a suitable environment for one agent*
  - The obstacles are to complicated for the algorithm.
  - The algorithm is not sufficient for items placed to close, which results in collisions.
  - The environment is too large or too small for the algorithm to handle.
- *Implement a single agent using a reinforcement learning algorithm*
  - The policy chosen for the agent to follow does not explore enough/exploit enough to find the optimal path. Proactive measure: Create a function now that computes the convergence rate for different policies, then choose the policy with the quickest rate.
  - The method of representing the states of the agent is poorly chosen.
  - The available actions are not enough to avoid collisions.
  - The rewards chosen are not optimal for training the agent.
- *Implement two agents using a reinforcement learning algorithm*
  - The algorithm developed for a single agent is insufficient for dealing with more than one. Then, the algorithm – including, action-value function, states, actions, rewards – might need to be modified greatly. Proactive measure: Consider the case of multiple agents from the beginning, adjusting the algorithm after that.
- *Implement multiple agents using a reinforcement learning algorithm*
  - As for the case with two agents, the already developed algorithm will probably need to be improved. Proactive measure: Make plans for incorporating a neural network in the algorithm.
- *Implement a neural network*

  - Neural networks are complicated to understand and implement and therefore more time consuming than anticipated. Proactive measure: Finish the implementation of two agents early and set aside a lot of time for researching and studying neural networks.

- *Integrate the reinforcement learning algorithm with neural networks*
  - Integrating the neural network with the reinforcement algorithm is complex. Thus a lot of code may need to be rewritten. Proactive measure: Contact the supervisors now and ask them about the best way to tackle this in order to have to rewrite as little code as possible.