CSC 36000: Modern Distributed Computing *NextGen with AI Agents*

Written Assignment #1

Total Points: 100 (+10 Bonus Points)

Instructions

Please answer the following questions to the best of your ability by writing them manually on a paper. Assignment submission is to be done by taking all photos of your results, making sure you are uploading all the photos and then upload them on Brightspace. Show all your work step-by-step for calculation-based problems. Your explanations should be clear and concise, demonstrating your understanding of the concepts from the lectures. The bonus question at the end is optional and you will not be penalized for not answering it.

Academic Integrity Statement (2 points)

I, **Write your Student Name here replacing this comment**, having Student ID **Write your Student ID here replacing this comment**, fully support this courses CSC3600's AI policy and the City College of New York's policy on Academic Honesty, which states the following in the Syllabus Document under GenAI policy and University policy

Note: You have to write the above academic integrity statement on a Paper manually on top of the manual writing components of this writing assignment. You will take photos of the writeup that you wrote manually and you will upload those photos showcasing your homework result. It is very important to manually write your homework as instructed for all questions.

Problem 1: Distributed System Models (20 points)

Consider a system for coordinating a fleet of autonomous delivery drones in a dense urban environment. The drones must communicate with each other and a central server to avoid collisions and optimize delivery routes.

a) (12 points) Classify this system using the three core properties discussed in the lectures. For each property, choose the most appropriate model and provide a detailed justification for your choice.

- **Time Model:** Synchronous or Asynchronous?
- **Failure Model:** Benign or Byzantine?
- **Communication Model:** Shared Memory or Message Passing?

b) (8 points) The FLP Impossibility Result states that consensus cannot be guaranteed in a fully asynchronous system if even one process might fail. Explain the practical implications of this theorem for your drone coordination system. What kind of guarantees can you *not* make, and what strategies might you employ to mitigate this issue in the real world?

Problem 2: Reliability and Algorithmic Capabilities in Modern Distributed Computing (23 points)

a) (6 points) Describe Byzantine Failure Tolerance while elaborating how it addresses Byzantine failures.

b) (3 points) If there are m Byzantine failures, how many replicas are needed to obtain the correct message?

c) (5 points) If a multi-agent system consists of M agents, and each agent can choose from a set of N distinct actions, what is the size of the joint action space (i.e., the total number of unique combinations of actions the agents can take in a single timestep)?

d) (5 points) Are classification algorithms sufficient for decision making in the real world with Distributed AI Agents? If so, why? If not, then why not?

e) (4 points) Write 3 strengths and 1 weakness of JAX based programming.

Problem 3: Multi-Agent Algorithms in Distributed Computing (20 points)

a) (5 points) Write 4 points of differences between the QMIX and the VDN algorithms.

b) (5 points) What is the difference between Q Learning and Independent Q Learning algorithms? Explain with examples having requisite number of agents as appropriate for the corresponding algorithms.

c) (5 points) Machine learning is often characterized as an optimization problem. Write down the general mathematical form of a supervised learning problem using a loss function. How does the objective of this optimization problem change or become more complex in a distributed system?

d) (5 points) Can shared memory models lead to race conditions? If so why? If not, why?

Problem 4: Q-Learning and Multi-Agent Coordination (35 points)

An AI agent is navigating a simple 1x4 grid world with states S0, S1, S2, and S3. The agent can only move 'Left' or 'Right'. Moving into a wall keeps the agent in the same state.

- **States:** {S0, S1, S2, S3}
- **Actions:** {Left, Right}
- **Rewards:**
  - Reaching S0 gives a reward of -10 (a penalty).
  - Reaching S3 gives a reward of +50.
  - All other moves give a reward of -1.
- **Learning Parameters:** Learning Rate $\alpha = 0.5$, Discount Factor $\gamma = 0.9$.
- **Initial Q-Table:** All values are 0.

a) (15 points) The agent starts in state S1. It performs the action 'Right', landing in S2. Using the Bellman Equation, calculate the new Q-value for the state-action pair $(S1, \text{Right})$. Show your work. The Bellman Equation is:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

b) (10 points) Now, assume the agent is reset to state S2. It performs the action 'Right', landing in S3. Calculate the new Q-value for $(S2, \text{Right})$. For this calculation, assume the Q-table is still filled with zeros.

c) (10 points) Imagine we now have two agents in this grid world. In Independent Q-Learning (IQL), each agent would learn its own Q-table as if it were acting alone. How does the objective of a Value Decomposition Network (VDN) differ from IQL? Write down the formula for the total Q-value (Q_{tot}) in VDN and explain in one sentence what it encourages the agents to do that IQL does not.

Bonus Question: Multi-Agent Algorithms for Distributed Systems (10 points)

a) (3 points) Would a q-learning algorithm or a multi-agent Q-learning algorithm be most useful for distributed systems and why?

b) (7 points) What are the weaknesses of Q-learning, IQL, and Centralized Multi-agent Q-learning (e.g. QMIX, VDN)