AI Economist: An Extension with Irrational Agents

Daniela Shuman

ABSTRACT

Reinforcement Learning has been recently used in Mechanism Design to simulate contexts with multiple players. The AI Economist, written in 2022, developed a Gather-Trade-Build economy simulation using a two-level reinforcement learning mechanism to determine an optimal tax policy [9]. While this model found the Saez result for the optimal tax value, the model does assume all agents are rational. This paper explores the use of quantal response equilibrium to explore irrationality within an economic simulation. First, the paper explores other irrational agents in both economic and reinforcement learning contexts. Then this paper explores using quantal response equilibrium, particularly the logit model to choose the optimal action in a well-known reinforcement learning environment. Finally, this paper explores the use of quantal response equilibrium in the original AI Economist with a Gather Trade Build Economy context.

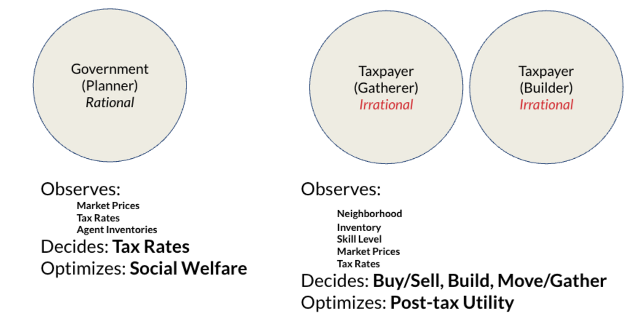

Figure 1: Visual Description of the AI Economist Multi-Agent System

1 INTRODUCTION

Policy decision making has had a hard time predicting optimal policies on the generational time scale due to the difficulty of obtaining such data. Typical prediction models (i.e. autoregressive or machine learning) tend to be inaccurate on the generational time scale due to this lack of data. Economists use Mechanism Design to consider the theoretical impacts of a policy on typical economic indicators (i.e. income, health, welfare, etc.). Mechanism Design is a framework that introduces a set of agents with different sets of incentives and uses this to determine the optimal rules for game theoretic problems.

AI is becoming a more powerful tool for policy-decision making. Particularly, the application of AI in mechanism design is becoming more popular as a tool to simulate societal behavior on the longer-term time scale. Reinforcement learning has been one such manifestation of models that have been used for mechanism design. Given reinforcement learning is an unsupervised mechanism, the above solution offers an alternative to using data. If we can model what a real economy looks like, we can simulate into the future to observe the effects of different tax policies on the outcomes we care about.

In a recent paper pioneered by SalesForce, the AI Economist uses two-level reinforcement learning to model the incentive structures and possible actions of the 1) Governing Body; and 2) Citizens [9]. This paper simulates a simple economy: Gather-Trade-Build economy. There are two types of citizens: Gatherers and the Builders. In this economy, each "citizen" has a different set of actions that they can take, their revealed preferences for their optimal outcomes, and their actual optimal outcomes. I go into detail about this in the methodology section.

The results found the Saez model, which demonstrate alignment with the theoretical economic community, in a one-agent system. In a multi-agent system, the AI Economist improves the utilitarian social welfare and the trade-off between equality and productivity over the baseline set by economic theory.

In this paper, we explore other papers that have evaluated the presence of irrationality in economic markets and in reinforcement learning in mechanism design. This paper also explores the converge of the Logit Model as a method to model irrationality of agents in a very simple Reinforcement Learning Mechanism design model. This paper found that the irrational setting did converge on a reward lower than the optimal, but only by 5-10%. Further, in the irrational setting, the agents were able to stabilize quickly (after approximately 400 iterations) to a cumulative reward. These results are useful steps in order to build confidence in applying directly to the AI Economist.

2 LITERATURE REVIEW

2.1 Irrationality in Economic Theory

Irrationality in economics is not frequently explored. Published in the International Economics Review, Mandler explored the necessary criteria for including irrational agents in economic markets [4]. Using economic theory, he questioned how robust economic models are to irrational agents, both in the context of games and economic mechanisms. It discusses the impact of irrational agents on Pareto efficiency in competitive equilibria, noting that the presence of even one irrational agent can lead to inefficiency. Indeed, mechanisms are sensitive to the presence of irrational agents. What he found was that there are robust alternatives with attractive welfare properties, particularly that the presence of irrational agents improves the welfare of rational agents. This makes intuitive sense, if a rational agent exists and is able to game given an irrational agent (one that choose from a uniformly distributed set of options), then there would be less competition amongst the agents to make the optimal choice, and the outcomes would be more skewed in the final distribution towards the rational agent. In other words, the rational agent would be more able to game the system than the irrational agents [4].

While the Mandler paper explored the presence of irrational agents in games, Rubbini (2023) explored mechanism design with irrational agents [6]. He found that relaxing the rationality assumption still requires Bayesian Incentive Compatibility to be met by the social choice function. Bayesian Incentive Compatibility means that every player can achieve their optimal outcome given they reveal their true preferences [3].

In the case of the AI Economist, it is unclear whether the social choice function itself is Bayesian Incentive Compatible. However, other papers have demonstrated that multi-agent reinforcement learning simulations, especially in the context of social welfare, tend to meet the criteria for Bayesian Incentive Compatibility [8]. When applying the AI Economist social choice function in this context, I assume that every player can maximize their true outcome by revealing their true preferences, largely because their true preferences are already known. Every tax payer wants to maximize their post-tax utility and the government agent wants to optimize social welfare. If taxpayers revealed preferences are different, this may end up hurting their post-tax utility. A more rigorous proof is required, which is what I discuss in the future work section. The AI economist paper does address strategic gaming and their results still held in the presence of this gaming. If we remove gaming artificially (as in force the revealed preferences to be the true preferences), we could artificially force the BIC characteristic to be the case. While I don't force the BIC characteristic in this paper, one could consider doing this in future work.

Rubbini found that the presence of both Rational and Irrational Agents can still achieve many results found in typical economic mechanisms. These economic mechanisms included the Myerson-Satterthwaite's Impossibility Theorem [5], which found that efficient bilateral trade is impossible in the presence of private information. This implies the necessity of BIC. The results also extended to the revenue equivalence theorem which showed that the necessity of BIC for implementation in Bayesian Naive Equilibrium, and allows the extension to all Weak Solution Consistency concepts. Thus, it is feasible to assume that the multi-reinforcement learning agent would still simulate the economy and return the optimal result found by existing economic theory.

2.2 Irrationality in Reinforcement Learning

Quantal Response Equilibrium has been specifically used in reinforcement learning to model irrational agents. Sokota et. al. (2022) [7] explores its applications in two-player zero-sum games, specifically in the context of reinforcement learning and quantal response equilibria (QRE). The paper also introduced a novel algorithm called Magnetic Mirror Descent (MMD) to solve two-player zero-sum games. The main advantage is that the algorithm is incentivized towards specific types of solutions and converges exponentially fast towards quantal response equilibriums. While this paper explored using MMD in multi-agent deep reinforcement learning algorithms for games such as 3x3 Dark Hez and Phantom Tic-Tac-Toe, the zero-sum nature of these results may extend to a social choice function with different types of tax payers. The authors claim that the theoretical and empirical results highlight MMD's potential as a versatile approach in the context of two-player zero-sum games.

Zhou and Xu (2019) [10] propose an algorithm to address the irrationality of cooperating agents caused by mechanical failures or sensor flaws in the context of electrical engineering. It is possible for two agents to corroborate in an economy, so an exploration as such is useful. The paper introduces a two-variable evaluation system (Cooperation Index and Competitive Flag) to quantify the cooperative/competitive level of the co-player in a continuous domain. The algorithm combines elements from Independent Learner (IL), Joint Action Learner (JAL), Nash Q-learning, and Friend-or-Foe (FoF) to handle dynamic changes in cooperation and competition. The contribution of the paper includes the introduction of the two novel parameters, the development of an online deep reinforcement learning algorithm to handle non-stationarity, and the extension of the Friend-or-Foe algorithm to deep reinforcement learning with the two-variable evaluation system. The proposed algorithm is claimed to achieve higher payoff compared to those assuming the co-player to be fully cooperative or competitive.

2.3 AI Economist Summary

The methodology in the AI Economist Paper is such that each actor learns a behavioral policy π to maximize its objective. The taxpayers begin under the conditions of neighborhood, inventory, skill level, market price, and tax rate. The social planner's inputs are the market price, tax rates, and agent inventories. The taxpayers decide on the behavioral policy by deciding whether or not they should move, buy/sell, or build. The taxpayers are deciding on post-tax utility. The social planner's behavioral policy is the tax rates. They are optimizing over social welfare.

The AI Economist currently achieves the following:

(1) In one-step economies, the AI Economist recovers the theoretically optimal tax policy derived by Saez (31). This demonstrates that the use of two-level RL is sound.
(2) In Gather-Trade-Build economies, tax policies discovered by the AI Economist provide a substantial improvement in social welfare for two different definitions of social welfare and in various spatial world layouts; e.g., in the Open-Quadrant world with four

agents, utilitarian social welfare increases by 8%, and the trade-off between equality and productivity increases by 12% over the prominent Saez tax framework (31). *In a Carbon Tax Framework, this Gather-Trade-Build economy may be questioned. We would need to add in further components to the economy.*

(3) In particular, AI social planners improve social welfare despite strategic behavior by AI agents seeking to lower their tax burden.

(4) AI-driven tax policies improve social welfare by using different kinds of tax schedules than baseline policies from economic theory. This demonstrates that analytical methods fail to account for all of the relevant aspects of an economy, while AI techniques do not require simplifying assumptions.

(5) This work also gives economic insights: It shows that the well-established Saez tax model, while optimal in a static economy, is suboptimal in dynamic economies where it fails to account for interactions between agents. Our framework enables us to precisely quantify behavioral responses and agent interactions.

3 METHODOLOGY

3.1 Quantal Response Equilibrium, Logit Model

The Logit model is the most widely used model to represent quantal response equilibrium (QRE). For each type of agent in the model i, there is a set of possible actions A_i, a set of utility functions $u_i : a_i \longrightarrow \mathbb{R}^+$. There also exists some noise β which is a hyperparameter in the model. QRE is a discrete probability distribution over a set of possible actions where the action that is the most optimal is chosen the majority of the time.

$$P(a_i) = \frac{e^{\beta u_i(a_i)}}{\sum_{b_i \in A_i} e^{\beta u_i(b_i)}}$$

A rational agent would have the following distribution:

$$P(a_i) = \{a_i = argmax_{b_i \in A_i} u_i(b_i) : 100\%\}$$

Irrational agents are typically modelled by a uniform distribution as follows:

$$a_i \sim \mathrm{U}(A_i), P(a_i) = \frac{1}{|A_i|}$$

The optimal action is $a_i^* = a : max\{u_i(a)\}$, or the action that maximizes the utility function. To illustrate how the optimal action is chosen more often with a simple example, we can observe in Figure 2 that the optimal action (which has utility 4.0) is chosen more times than the inoptimal actions, but there is still some non-zero probability that the other actions are chosen. In the rational agents case, the optimal action would be chosen 100% of the time.

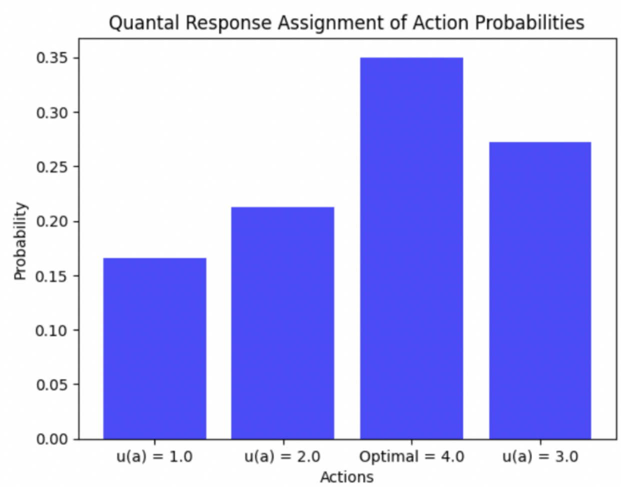

Figure 2: QRE Example 1

If the optimal action has a significantly larger utility value than the other the other actions, then the probability of choosing said action increases (exemplified by Figure 3).

We can explore the relationship between the probability of choosing the action and the more optimal the action becomes. The relationship plateaus near 100% as the optimal has a higher utility than the other action options in the context of 4 actions.

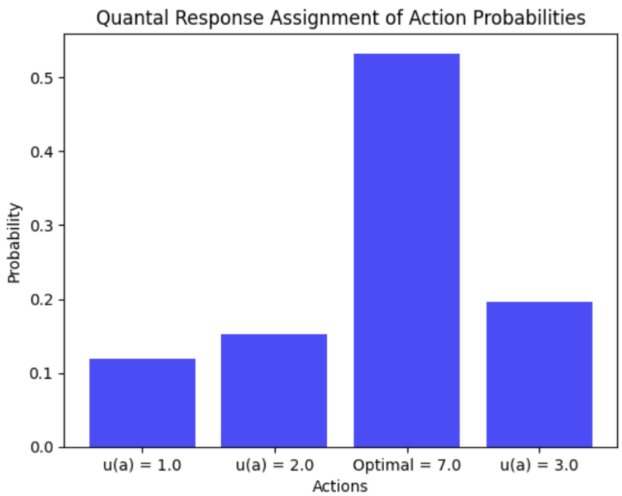

Figure 3: QRE Example 2

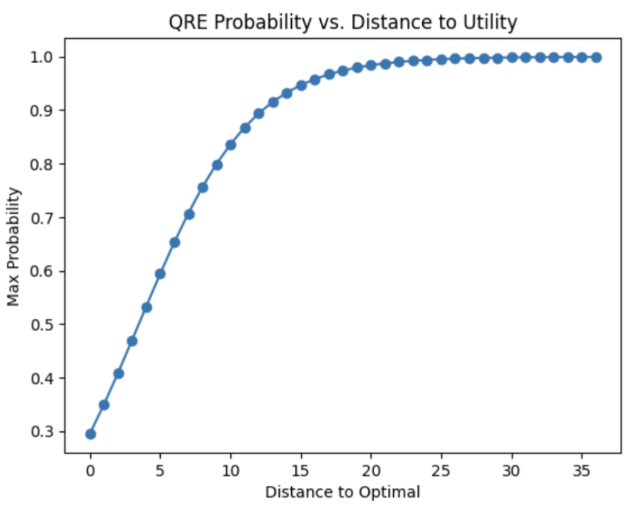

Figure 4: QRE Optimal Probability as Action becomes more optimal

4 RESULTS

OpenAI Gym's Taxi-v3 is a classic reinforcement learning environment where an agent controls a taxi navigating a grid-world. The goal is to pick up a passenger from one location and drop them off at another while optimizing for time and avoiding illegal pick-ups and drop-offs. The environment provides a discrete action space representing the taxi's movements and a discrete state space indicating the locations of the passenger, destination, and the taxi itself. The agent receives rewards or penalties based on its actions and the achieved task, making it a popular choice for learning and testing RL algorithms. I used this environment to explore the presence of irrational agents and how the reward converges over time.

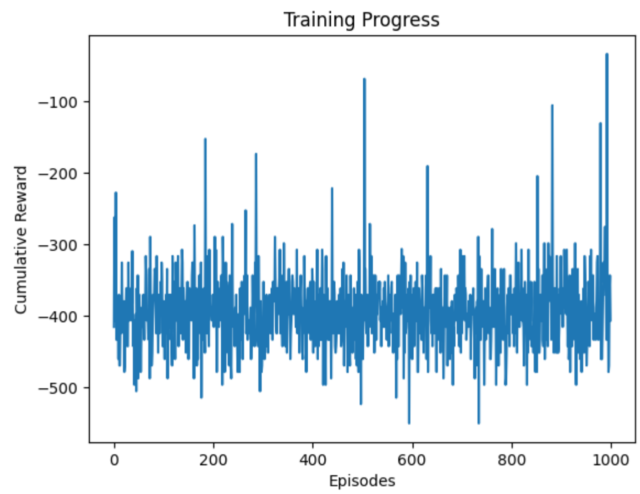

Figure 5: OpenAI Taxi-v3 with Irrational Agents, Uniform Irrationality

The above figure displays the cumulative reward when choosing the action uniformally at random. We see that there is little convergence after 1000 iterations on a cumulative reward.

With the QRE, we were able to observe how the cumulative reward converges after apprxoiamtely 400 iterations. Since the optimal utility function is chosen more often, the agents are more likely to converge on a higher utility. With the rational agents, the cumulative reward converges at a higher reward than the irrational agents with QRE.

The above results suggest that the same result may be achieved with irrational agents as with irrational agents when choosing to model irrationality with QRE. This implies that a tax policy may reach practical optimally after a period of time in an irrational setting.

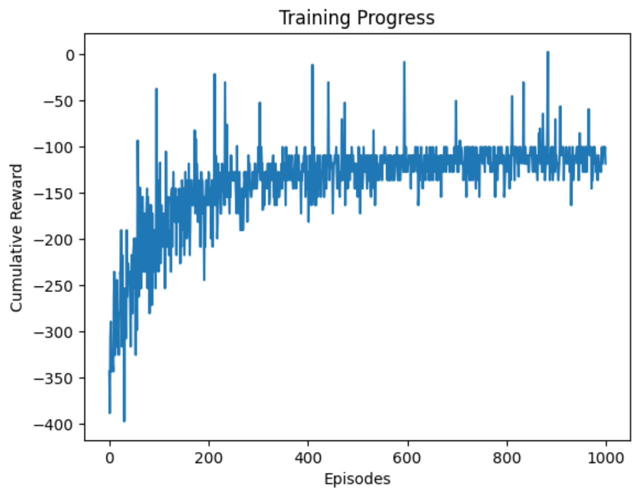

Figure 6: OpenAI Taxi-v3 with Irrational Agents, QRE Irrationality

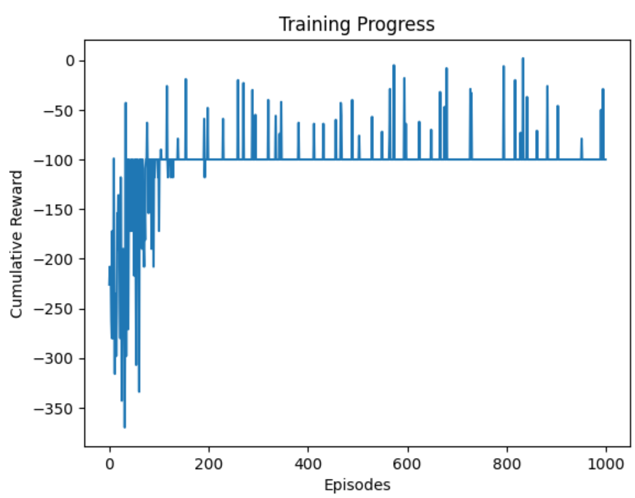

Figure 7: OpenAI Taxi-v3 with Irrational Agents, Rational Agents

5 CONCLUSION + FUTURE WORK

This paper was an exploration of applying irrational agents to the two-level reinforcement learning. This paper investigates the introduction of irrationality into the economic simulation by employing the quantal response equilibrium using the logit model. Initially, the study explores irrational agents in economic and reinforcement learning settings. Subsequently, it delves into the application of quantal response equilibrium, specifically the logit model, to select optimal actions within a well-known reinforcement learning environment. Lastly, the paper examines the integration of quantal response equilibrium into the original AI Economist framework within the context of the Gather-Trade-Build Economy.

Based on the Sokota et. al. paper [7], an exploration into using MMD instead of Stochastic Gradient Descent typically used in reinforcement learning problems may result in the findings of the results of the paper quicker than the original paper did under the presence of irrational agents. Extending the results of this paper by using different iterative algorithm for optimizing an objective function could be useful and insightful.

Exploring other functions for irrationality may be useful. Irrationality has been explored in financial markets with the psychological lens [2] and in Game Theory more broadly [1].

The Rubbini paper found that Bayesian Incentive Compatibility is necessary in order to achieve optimal results in the presence of irrational agents. However, it is not directly obvious that the AI Economic social choice function is Bayesian Incentive Compatible. This suggests that a proof is necessary in order to fully apply the findings of the Rubbini paper to this paper's exploration.

Need to model the utility function of each taxpayer specifically with the probability to observe the realized action chosen

Modelling a more complex economic landscape would be useful practically. A Gather-Trade-Build economy does not engage with the complexities of a modern economy, where income distribution is continuous rather than discrete, and there are more ways to collaborate strategically between taxpayers in society. Further, I would like to expand the experimental results to include different kinds of utilities. It is not clearly explored in the above paper what kinds of actions are more or less clear to different taxpayers. Potentially the optimal choice for the set of actions is obvious and irrationality is realized in other ways.

While I was unable to run the AI Economist due to issues with the package, I hope to send an updated summary of my results to the teaching team when the issue with the package is resolved (currently in communication with original authors of the AI Economist paper).

6 ACKNOWLEDGEMENTS

I really enjoyed working on this project and taking this class. Being able to explore something I care about (making the world a better place through theoretical CS) and combining my recent professional experience in macroeconomic theory was a very rewarding experience. I hope to continue to work on this project (as mentioned above). I'm very grateful to the teaching staff, particularly Prof. Yiling Chen for her support on this project throughout the semester!

REFERENCES

[1] How game theory explains 'irrational' behavior | MIT Sloan — mitsloan.mit.edu. https://mitsloan.mit.edu/ideas-made-to-matter/how-game-theory-explains-irrational-behavior, 2022.
[2] Dhesi. Modelling and measuring the irrational behaviour of agents in financial markets: Discovering the psychological soliton.
[3] M. O. Jackson. Mechanism theory. https://web.stanford.edu/~jacksonm/mechtheo.pdf.
[4] M. Mandler. Irrationality-proofness: Markets versus games. *International Economic Review*, 55(2):443–458, 2014.
[5] R. B. MYERSON and M. A. SATTERTHWAITE. Efficient mechanisms for bilateral training. https://www.cs.princeton.edu/courses/archive/spr09/cos444/papers/myerson_satterthwaite83.pdf, 1981.
[6] G. Rubbini. Mechanism design without rational expectations. *arXiv preprint arXiv:2305.07472*, 2023.
[7] S. Sokota, R. D'Orazio, J. Z. Kolter, N. Loizou, M. Lanctot, I. Mitliagkas, N. Brown, and C. Kroer. A unified approach to reinforcement learning, quantal response equilibria, and two-player zero-sum games, 2023.
[8] L. Xia. Designing social choice mechanisms using machine learning. https://www.cs.rpi.edu/~xial/Files/XiaAAMAS13.pdf, 2013.
[9] S. Zheng, A. Trott, S. Srinivasa, D. C. Parkes, and R. Socher. The ai economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances*, 8(18):eabk2607, 2022.
[10] Z. Zhou and H. Xu. Deep reinforcement learning based intelligent decision making for two-player sequential game with uncertain irrational player. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 9–15, 2019.

7 APPENDIX

Below contains the reinforcement learning code for including the irrational agents.

```
import numpy as np
import random
import gym
import matplotlib.pyplot as plt

class IrrationalQLearner:
    def __init__(self, num_actions):
        self.num_actions = num_actions
        self.q_values = np.zeros((500, num_actions))

    def choose_action(self):
        # Simple irrational behavior: choose a random action
        return random.randint(0, self.num_actions - 1)

    def update_q_values(self, state, action, reward):
        # Q-learning update rule (simplified)
        self.q_values[state][action] += 0.1 * (reward - self.q_values[state][action])

def run_simulation():
    env = gym.make('Taxi-v3')
    num_actions = env.action_space.n

    # Agents
    higher_level_agent = IrrationalQLearner(num_actions)
    lower_level_agent = IrrationalQLearner(num_actions)

    num_episodes = 1000
```

```python
    cumulative_rewards = []

    for episode in range(num_episodes):
        state = env.reset()

        # Higher-level agent decides on a high-level action
        higher_level_action = higher_level_agent.choose_action()

        total_reward = 0

        for _ in range(100):  # Maximum of 100 time steps per episode
            # Lower-level agent decides on a low-level action
            lower_level_action = lower_level_agent.choose_action()

            # Execute the combined action in the environment
            combined_action = lower_level_action  # Simplified for this example
            next_state, reward, done, _ = env.step(combined_action)

            # Update Q-values of both agents
            higher_level_agent.update_q_values(state, higher_level_action, reward)
            lower_level_agent.update_q_values(state, lower_level_action, reward)

            state = next_state
            total_reward += reward

            if done:
                break

        cumulative_rewards.append(total_reward)

    env.close()

    # Plotting
    plt.plot(cumulative_rewards)
    plt.xlabel('Episodes')
    plt.ylabel('Cumulative Reward')
    plt.title('Training Progress')
    plt.show()

if __name__ == "__main__":
    run_simulation()
```