ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Final Project Report

ETH Zürich

Complex Social Systems: Modeling Agents, Learning, and Games

Learning to Trade: Examining Agent Behaviour in Markets

Authors:
Les truites saumonées
Florian Abeillon
Sverrir Arnórsson
Yilin Huang

Professor:
Prof. Dirk Helbing

Teaching Team:
Dr. Nino Antulov-Fantulin
Thomas Asikis

December 5, 2021

1 Abstract

This project shows a proof of concept of using a Reinforcement Learning (RL) approach to create a market environment with agents that use Q-learning to update their knowledge and influence future actions. In addition to creating a market, this project identifies key parameters that influence the behaviour of the market's agents, and seek to identify the effect these parameters have. The key parameters this work focuses on are the agent's memory (learning rate, α), risk tolerance (discount factor, γ), and curiosity (ϵ-greedy function). Each seller is required to learn the quantity of good to produce and price to sell it for given a set production cost, while each buyer learns what quantity it should purchase given a set budget. Both buyers and sellers have a unique reward function to encourage them to maximize their profit and goods acquired, respectively. The impact of altering these key parameters are then compared to a baseline to determine how important the parameters are to this agent-based financial model.

Keywords: Reinforcement Learning, Q-learning, Agent-Based Modelling, Financial Market

Agreement for free-download

We hereby agree to make our source code for this project freely available for download at `https://github.com/florian-abeillon/debunking_invisible_hand`. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Florian Abeillon, Sverrir Arnórsson, Yilin Huang

Contents

List of Figures

List of Tables

2 Introduction and Motivation

Over the past two decades, there has been significant interest and work in replicating and predicting financial markets. The field itself gained significant traction with LeBaron's Sante Fe artificial stock market [1], and since then, much work has been done to apply bottom-up models of the financial market. Models like this start from first principles of the agents' behaviour, and are known as agent-based models (ABM). Many of these models propose heterogeneous agents with learning and optimisation capabilities, however, these systems often have significant complexity and require considerable computing power [2].

This project aims to *create a faithful agent-based representation of a market and determine the most important parameters that affect it.* The market's model is based on a Markov Decision Process (MDP) adjusted to suite a financial application, while the agents learn through updating what they know (ie. results/reward from a particular state). This learning process emulates Reinforcement Learning (RL). In this way, we want to show that the so-called "Invisible Hand" is not a magical phenomenon, but rather a consequence of individual-level dynamics within a complex system.

Reinforcement learning is particularly suited to this financial market models since it allows the agents to learn the strategies to achieve their goals. Unlike other branches of machine learning such as supervised or semi-supervised learning, RL is not constrained by the need for training/validation datasets. This way, the agents are able to explore different strategies [3].

2.1 Why Intelligent Agents?

To understand the motivation behind using trained agents (ie. intelligent agents), it is beneficial to first understand what the results of a non-intelligent agent would be. In computational financial markets, the term Zero-Intelligence Agent (ZIA) is used for a trader that makes purchases and sales randomly, based only on minimal constraints.

There is evidence that ZIAs are still able to trade at a relatively effective level, which leads to the conclusion that the primary driver of market efficiency may be simple laws that govern the market prices and order flows that are imposed on agents, and that the agents' learning, intelligence, and profit motivation may not have such a large factor on performance [4] [5].

The simplicity of these ZIAs are useful to researchers in understanding model behaviour, and as such, they are often used to explain complex market behaviours, as demonstrated in [4] and [6]. Models that attempt to capture intelligent, strategic behaviour, on the other hand, often have a myriad of parameters, making it difficult to pinpoint the cause of a particular effect [7].

The problem with these systems, however, is that since they are based on randomized actions, there is no learning involved, and thus no method to analyze or extrapolate their actions. As such, ZIAs are an important baseline model that computational financial systems compare against, but are rarely (ie. never) used as a development model since they lack analytical tractability. In addition, while ZIAs

are able to reach an equilibrium in a market, they will perform random actions if they are exposed to a new market scenario – while in reality, humans (or intelligent agents) will have learned. Thus over several temporal iterations, the results of ZIAs are not likely to be fully reflective of real-life market scenarios [7].

As such, this project aims to use agents that learn over time, yet are only constrained by key parameters, their theory described in Section 4, and the impact of the parameters on the model described in Section 5.

2.2 Related Works

Significant interest has been devoted to modelling financial systems using agent-based models, whether in a single-auction (simple market) or double-auction (stock exchange, cryptocurrency) set-up. As such, there is a substantial body of work that supports this proof-of-concept.

LeBaron's work on the Santa Fe artificial stock market [1] in 2002 initiated significant interest in the capabilities of reinforcement learning systems to model a financial system that replicated real-world stock markets. In the ensuing years as computation power became greater, researchers started to focus on giving agents a wider range of trading options. Research such as Boer's on an agent-based framework in artificial financial markets shows that instead of having restrictive assumptions, allowing agents to try out a larger range of trading strategies can result in more efficient markets [8].

Cristelli provides a comprehensive overview of a wide-range of financial models developed over time, and how quantitative assessments of economic and financial issues are approached [9].

Traditional attempts at modelling a market have focused on heterogeneous agents that have static rules. The issue with these models, however, has always been that they are rather difficult to validate, and parameters can be altered by the researcher. To reduce the affects of these issues, it has been suggested that researchers should put these parameters under evolutionary control — this way, the system can be validated without calling into question the setting of certain parameters [10].

The trend towards agents that are not bound by simply static rules has garnered more attention in the past decade with the increase in computational power as well as machine learning techniques. As early as 2004, Lo proposed an adaptive market where agents adapt their strategies depending on their environment [11].

In the late 2000s, work such as that of Rutkauskas and Ramanauskas introduced the concept of applying Reinforcement Learning to the field of computational financial markets, as described in [12]. Rutkauskas and Ramanauskas employ the use of Q-learning (developed in 1989 by Watkins, as described in [13]) and demonstrated that the strategies developed by the agents exhibited rational economic behaviours and optimization techniques [3].

3 Market System

The model is based on single-auction market system with two types of agents (seller and buyer), and the environment they conduct transactions in. Algorithm 1 shows the workflow of what happens when the market opens, and is represented by the green arrow in the system overview in Figure 1. The entirety of Algorithm 1 represents one round. To train the agents, several thousand rounds (a "game") are run so that they can learn to better reach their goal. All the tests conducted in this project, described in Section 5 were run with at least 25,000 rounds.

Shuffle the buyer order to consider them in a random order;
foreach *buyer* **do**
  Select a random number of sellers k so that $k \in \{1, ..., n_{\text{sellers}}\}$;
  foreach *selected seller* **do**
    Get price and quantity from seller;
    Get how much the buyer wants to buy, based on Q-table with $p = 1 - \epsilon$ or at random with $p = \epsilon$ [a];
    if *buyer wants to buy something* **then**
      Make seller sell
    end
    Make buyer learn based on transactions
  end
end
Make all sellers learn

Algorithm 1: Market interaction and agent learning process

[a] The selection of *epsilon* is described in section 4.5

4 Q-Learning and Model Parameters

Reinforcement learning, a branch of Machine Learning, involves the use of goal driven agents, a set of states S, and a set of actions A per state. The agent can perform any action $a \in A$ to transition from the possible states. By taking an action in a specific state, the agent receives a reward, with the agent's goal being to maximize its total reward. This is done by adding the maximum attainable reward from the future to the reward received for reaching its current state. This way, the future reward probability influences what action the agent should currently take [14]. This is represented mathematically in Equation 1.

$$Q(s_t, a_t) \leftarrow (1 - \alpha) Q(s_t, a_t) + \alpha \cdot \left(r_t + \gamma \cdot \max_a \left[Q(s_{t+1}, a) \right] \right) \tag{1}$$

Where α is the learning rate, in this project's case the agent's memory (described in Section 4.3), r_t is the reward (described in Section 4.2) and γ is the discount factor, in this case, an agent's tolerance to risk (described further in Section 4.4). The agent uses a greedy policy to increase its chances to reach an optimal balance of exploitation (choosing the seemingly reward-maximizing option) and exploration (discovering other options in the environment).

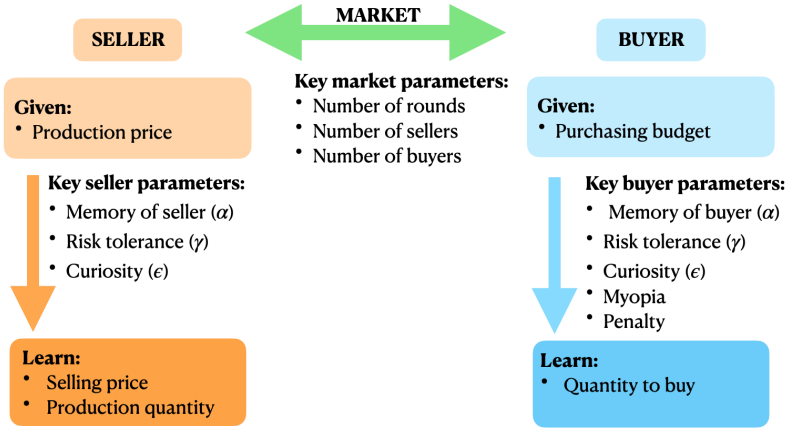


Figure 1: System overview of the market, including seller and buyer parameters and learning objectives.

4.1 Q-table

Each agent learns by updating the reward they receive based on a specific action. These values are stored in a Q-table, which is a table of all states and associated actions. Each agent has a different

4.1.1 Sellers Learning

The sellers' Q-table is set in a way that there is only one state, and several actions. The state is the generic market state, as it does not change over the time of the simulation. The actions are how much the seller decides to produce and the price that it decides to sell for. One could look at this as a table with one row where different combinations of quantity and price constitute the columns. The seller's learning goal is to decide on the optimal combination of selling price and production quantity.

4.1.2 Buyers Learning

For the buyers, the states (Q-table rows) are all the possible pairs of budget the buyer has left and the price they're offered the good at. The actions (Q-table columns) are all the possible quantities they could choose to buy. When learning, buyers are able to explore this space to determine what results in the maximal reward. The results of how the Q-table looks in this market system is shown in Figure 4.

4.2 Reward Function

The reward function for sellers is explained in Section 4.2.1, and that for buyers is explained in Section 4.2.2.

4.2.1 Rewarding Sellers

In this project, the goal of the sellers is to maximize their profit, which is set as:

$$R_{seller} = n_{sell} \cdot p_{sell} - n_{prod} \cdot p_{prod} \tag{2}$$

where p_{sell} is the selling price and p_{prod} is the production price. The reward function is then implemented as $Q_{seller} = P_{seller}$, where Q is the reward as explained in Section 4.

In Equation 2, the production price is a parameter set before the learning, ie. all sellers have a CHF 10,- production price. During the reinforcement learning, the sellers learn at what price to sell for (p_{sell}), as well as how many goods to produce (n_{prod}). The number of goods sold is determined by how many goods the buyers seek to purchase (something the buyers learn concurrently). This system is illustrated in Figure 1.

Sellers who make a positive profit are therefore given a positive reward and encouraged to repeat this action. When seller actions result in a loss, the reward function is negative and thus acts as a penalty.

4.2.2 Rewarding Buyers

In this system, buyers are treated as vendors rather than end-users. Substantively, this means the buyers' goal is to purchase the maximum number of goods rather than to purchase a desired n goods. For instance, an end-user may only desire 5 goods, and thus would not need (or want) to purchase more than 5 goods. A vendor, on the other hand, wishes to maximize the profit it gets from re-selling this good, and is therefore seeking to maximize the goods it can purchase for a suitable price. For this, the reward is directly correlated to the quantity purchased, ie. $R_{buyer} = n_{purchased}$.

This, however, is not the only consideration for the buyer. The buyer seeks to purchase more goods overall, and since sellers are concurrently learning their selling price, the price from the first seller may not be optimal for the buyer. As such, buyers learn to be patient and to not spend all their budget on the first vendor. This reward for patience is implemented as a"global reward" (as opposed to the previous, "local", reward).

After every market round (described in section 3), the buyer reviews the purchases made from vendors in the previous round. The reward for every purchase is described in Equation 3.

$$Q_{buyer}(b_{before\ purchase}, p_{offered}, n_{bought}) = (1 - m) \cdot r_{local} + m \cdot r_{global} \tag{3}$$

Where $b_{before\ purchase}$ is the budget before the buyer made the purchase, $p_{offered}$ is the price they were offered to buy at, n_{bought} is the number of goods they bought from this seller, r_{local} is the local reward and r_{global} is the global reward. These factors are then linked by the myopia, m, which is a measure of how much each the global, long-term reward is favoured over the local, short-term one. The local reward r_{local} is defined as

$$r_{local} = n_{bought\ from\ seller} - \mu \cdot b_{left\ after\ interaction} \tag{4}$$

where μ is a penalty for not spending all the budget, $n_{boughtfromseller}$ is the number of goods purchased from a specific seller, and $b_{leftafterinteraction}$ the budget left after interacting with this specific seller. The global reward r_{global} is defined as

$$r_{global} = n_{bought\ after\ this\ seller} - \mu \cdot b_{left\ after\ round} \tag{5}$$

which includes the same penalty as in Equation 4, $n_{bought\ after\ this\ seller}$ the number of goods purchased in the round after a specific seller, and $b_{left\ after\ round}$ is the budget left after interactions with all sellers for that round.

4.3 Memory (α)

The main purpose of implementing an intelligent agent is to ensure it is learning from its actions. How quickly the agents learn is defined by the rate newly acquired information replaces old information, α, which represents the agent's memory in this market system. If α is set to 0, then the Q-values in the Q-learning table are never updated, and the agent never learns. This does not mean, however, that a high learning rate is better. If the learning rate is too high, the agent only uses the most recent information, and ignores all prior collected knowledge [14]. This means, for each agent, an optimal memory value (α) exists, which is largely found through trial and error.

4.4 Risk Tolerance (γ)

The market agents aim to learn based on maximizing total rewards. As explained in Section 4, this maximization is a combination of current rewards and possible future rewards. The risk tolerance factor, γ (often called the "discount factor"), determines the present value of future rewards. If the risk tolerance is high, then the agent is more willing to take a risk and strive for a long-term reward rather than taking the immediate reward. This is set because an agent that only chooses to maximize immediate rewards may not end up exploring a substantial amount of the environment, and therefore may have a reduced total reward[15].

4.5 Curiosity (ϵ)

In a typical reward-maximizing model, an agent would consistently choose the action that yields the highest expected reward, known as a greedy policy. In such a scenario, there would be no room for exploration. Optimal learning is however achieved through a balance of exploitation and exploration. In this case, it is important the agent follows an ϵ-greedy policy, which we call the "curiosity" component. In ϵ-greedy, a number (ϵ) in the range $[0, 1]$ is selected prior to selecting an action. A random number is then generated: if it is larger than the set ϵ, the agent chooses the greedy action, and vice versa for a number smaller than ϵ [16].

For the market, both the sellers and the buyers have a curiosity component. The buyers' curiosity determines how often they "try" randomly new prices and production quantities, while the buyers' curiosity has them "try" random quantities to buy. The variation of this curiosity function over the rounds elapsed is based on a sigmoid function, as illustrated in Figure 2.

5 Model Parameter Calibration

To determine the effects of each parameter on the model, a baseline must be established. In Section 4, each parameter's function is described and motivated as to why it is necessary. The following sections focus on the baseline model parameters

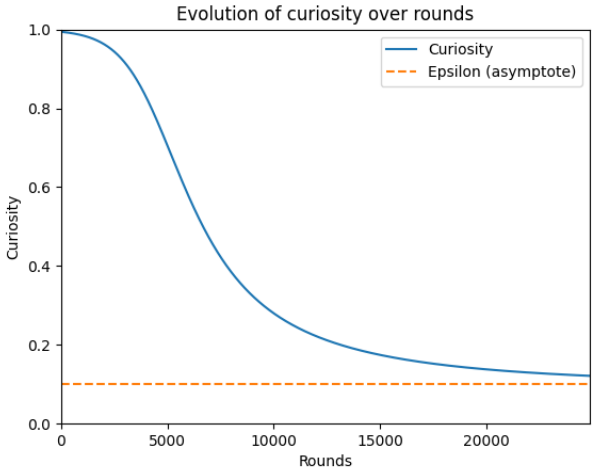


Figure 2: Variation of ϵ over the rounds in the simulation

and the effects on the seller and buyer behaviours (Section 5.1), and the effect of increasing/decreasing the parameters is described in Section 5.2.

5.1 Baseline Model Parameters

5.1.1 Sellers: Baseline Parameters

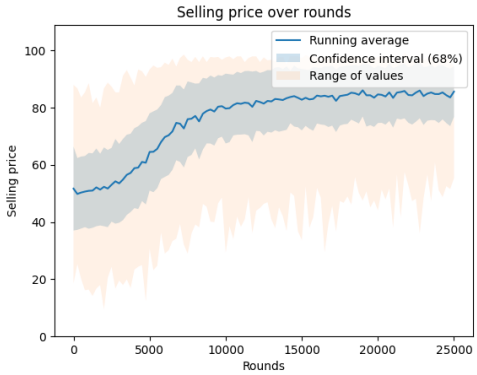
The baseline values for the sellers are shown in Table 1 and used to generate the trends shown in Figure 3.

Table 1: Seller baseline parameter values

Parameter	Value
Range of possible selling prices	[1,100]
Range of possible production quantities	[1,100]
Seller production price	10
Memory (α)	0.1
Risk tolerance (γ)	0.5
Curiosity (ϵ)	0.2

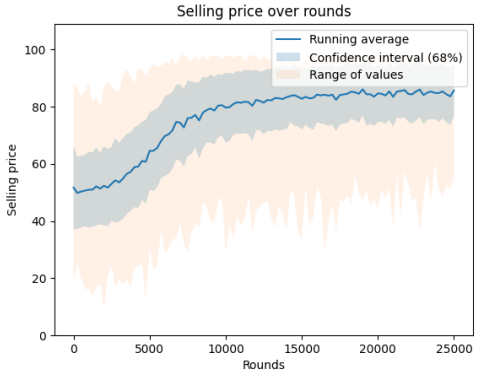
Note that the sellers initially produce a random number of goods, hence the average at 50 (Figure 3a), but learn to increase this amount, so as to eventually stabilize at around 85 goods produced. As this production quantity increases, they also learn to decrease their selling price, as shown in Figure 3c. Figure 3d shows that these actions are maximizing the profits earned by the seller, as they go from being in the -200 range to around 160. This is in accordance with the seller's goal of maximizing profit, as described in Section 4.2.1. While some of these actions may not seem intuitive based on human values, it is important to remember that a computational agent that is given a reward function may not seek to optimize with a method that is necessarily "sensible" to a human. This is due to a fundamental difference in how humans and computational agents learn and is described in further detail in Section 6.2.

Figure 4 shows the Q-table of the seller's learnings. Each cell corresponds to

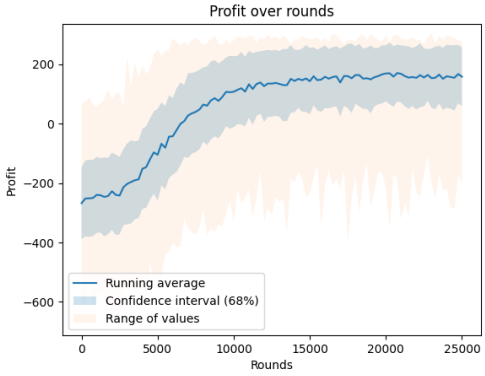


(a) Quantity of goods produced

(b) Quantity of goods sold



(c) Selling price



(d) Profit generated

Figure 3: Baseline results for the key parameters for the sellers (25,000 rounds)

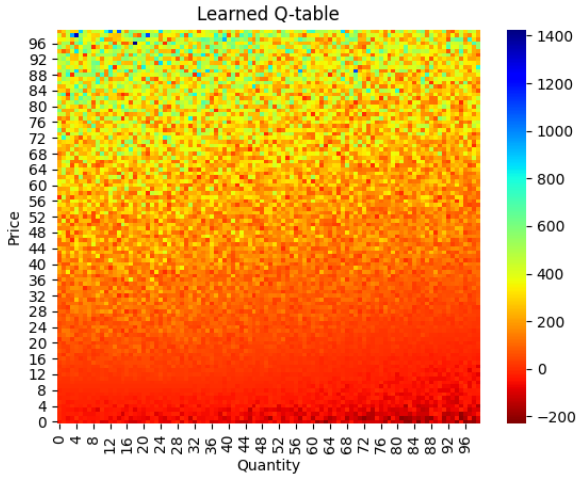


Figure 4: Seller's baseline Q-table

a pair (Quantity produced, Selling price). The negative values (here in red) are consequences of experienced losses, while positive values (here in colors other than red) are consequences of experienced profits. The higher the value (the more "blue"), the higher the experienced profit and the higher the chance this action is reward-maximizing.

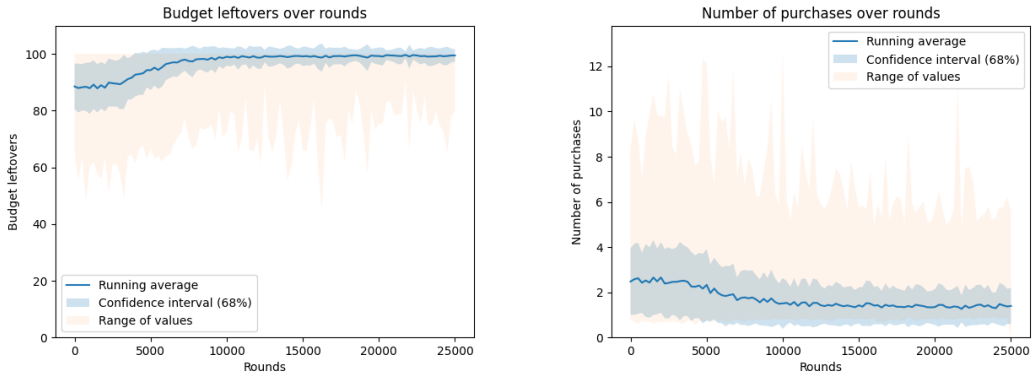
5.1.2 Buyers: Baseline Parameters

The baseline values for the buyers are shown in Table 2 and used to generate the trends shown in Figure 5.

Table 2: Buyer baseline parameter values

Parameter	Value
Range of possible budget allocations	[1,100]
Initial budget	100
Memory (α)	0.1
Risk tolerance (γ)	0.2
Curiosity (ϵ)	0.1
Myopia (Short-sightedness) factor	0.2
Penalty for having budget remaining	0.5

Note that the trends of the buyers correspond with that of the sellers actions. In particular, as the sellers learn to raise their prices at around 8,000 rounds (see Figure 3c), the buyers subsequently learn to buy less (see Figure 5b), but also learn to raise their budget allocation (see Figure 5a). Note that as buyers are more numerous than sellers, the latter can be profitable even if the individual number of buyers' purchases is low.



(a) Budget leftover

(b) Quantity of goods purchased

Figure 5: Baseline results from the key parameters for the buyers (25,000 rounds)

5.2 Effect of Parameters on Model

5.2.1 Sellers: Effect of Parameters

The effect of the parameters are shown in Table 3, in comparison to the baseline values stated in Table 1. The main conclusions are that production prices are the biggest factor in the parameters related to sellers along with curiosity. However, curiosity's impact is mainly in reducing learning, so it is not as relevant as production prices.

Table 3: Effect of altering the sellers' parameters

Parameter	General effect
Production Price	
High	Fewer purchases, less production, extremely high losses, Q-table slanted with positive in top-left, low in bottom-right, lower prices
Low	Higher profits, production quantity stagnant
Memory	
High	Negligible difference
Low	Q-table near zero, a bit higher quantities produced, otherwise negligible difference
Risk Tolerance	
High	Slightly lower profits, lower prices, and higher quantities
Low	Lower prices, similar profits,
Curiosity	
High	Significant difference: more purchases, lower prices, fewer quantities produced, more positive Q-table scores
Low	Lower profits, higher quantities produced

5.2.2 Buyers: Effect of Parameters

The effect of the parameters are shown in Table 4, in comparison to the baseline values stated in Table 2. What is interesting here is that curiosity does not have as big of an impact as with sellers, suggesting that they are not learning enough in the baseline.

Table 4: Effect of altering the buyers' parameters

Parameter	General effect
Budget	
High	Baseline model
Low	Fewer purchases, lower sales prices, less budget left, more negative Q-table scores, higher quantities produced, fewer sold quantities
Memory	
High	More variance in budget left for buyers, less quantity produced, lower profits, same quantities produced, lower selling prices
Low	More variance in budget left for buyers (but not as much of a difference as with high memory)
Risk tolerance	
High	Lower prices, slightly different profits
Low	Lower profits, higher prices, same prices
Curiosity	
High	Negligible, slightly higher lower profits
Low	Slightly higher quantities produced
Myopia	
High	Lower selling prices, substantially lower profits, quantity produced does not change
Low	Slightly lower profits (but not as much as with high myopia), higher quantities produced, slightly less sold quantity
Penalty	
High	Higher quantities produced, lower profits, slightly less sold quantity
Low	Fewer purchases, lower prices, lower profits, production quantity stagnant, very little sales overall

5.2.3 General Parameters

The effects of general model parameters can be seen in table 5. Their effects are more drastic than the buyers' and sellers' parameters, and is mostly in line with basic microeconomics, i.e. that more buyers result in higher prices and fewer buyers results in lower prices.

Table 5: Effect of altering the general parameters on the seller's and buyer's actions

General parameters	
Number of sellers	
High	Less budget left over, sellers run at a loss, q-table is over a smaller range, selling price is higher, each seller sells fewer products
Low	Unreliable, values vary too much
Number of buyers	
High	Higher quantities, higher prices, higher profits
Low	Less budget left over, less profit, lower prices

6 Improvements & Further Research

6.1 Using Dynamic Goals Instead of Static Goals

Currently, the agents in this project have static rules. This means that if a buyer has a set discount factor (or even, discount function), that factor will always apply throughout all iterations of the simulation. The majority of artificial markets have

also modelled behaviour using agents with a static set of rules [3]. There have been suggestions, such as those from Lo [11] that instead of static rules, agents would benefit from having an evolutionary approach. The idea is that individual agents adapt their strategies through trial and error, and strategies that garner more rewards dominate over time. In this way, it is not necessary for them to have initial strategies (ie. myopia - buy less at the beginning, buy more near the end), but rather they would learn how to change their strategies given their changing environment (ie. sellers could adapt to the pricing of their competitors).

6.2 Using Target Behaviour Instead of Rewards

Reinforcement learning is able to replicate, and in many instances, also supersede human performance, but all RL requires someone to manually specify a reward function (as we did with the buyers and sellers). However, it is often more intuitive to provide examples of target behaviour rather than designing what the reward should be.

Inverse Reinforcement Learning (IRL) algorithms use this as the basis of how they operate, and infer the reward based on demonstrated behaviour [17]. In some cases, we are able to give a relatively straightforward goal (ie. maximize profit), but it is often a black box what actions the agent will take to reach this goal. Since machines inherently do not share human values, actions may be taken that are inherently unpredictable to humans. by showing the desired behaviour, and having the agent infer the goal, we can better predict the foreseeable actions.

6.3 Creating a Dynamic Market Environment

In this current project iteration, the sellers' Q-table is one row, reflecting the fixed market environment. One could imagine that having a dynamic market environment where the number of buyers and sellers could vary between rounds would be an interesting next step. In this scenario, information (ie. number of buyers, seller prices) could be used as the states of the seller Q-table. This however, would increase the complexity of the model, which is slightly counter to what the initial goal of this project was — to create a representative market model and determining the impact of only the *most important* parameters.

7 Conclusions

This project shows a proof of concept for a market environment with reward-maximizing agents that learn to buy and sell through Q-learning. It also shows the important parameters that affect agent behaviour, notably the number of buyers/sellers, the buyers' budget, and the production price all have a strong effect on the agent behaviour. While this model demonstrates how parameters can affect a reinforcement learning market model, it has some drawbacks in that the buyers do not buy enough and are hard to incentivize; as such, it becomes hard for the sellers to generate profit. This project shows the parameters that are important, but further optimization is needed to ensure the model can better reflect real-world situations.

References

[1] B. LeBaron, "Building the santa fe artificial stock market," *Physica A*, pp. 1–20, 2002.

[2] M. Raberto, S. Cincotti, S. M. Focardi, and M. Marchesi, "Agent-based simulation of a financial market," *Physica A: Statistical Mechanics and its Applications*, vol. 299, no. 1, pp. 319–327, 2001, Application of Physics in Economic Modelling, ISSN: 0378-4371. DOI: https://doi.org/10.1016/S0378-4371(01)00312-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437101003120.

[3] M. Kvalvær and A. Bjerkøy, "Replicating financial markets using reinforcement learning; an agent based approach," M.S. thesis, NTNU, 2019.

[4] D. K. Gode and S. Sunder, "Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality," *Journal of political economy*, vol. 101, no. 1, pp. 119–137, 1993.

[5] J. D. Farmer, P. Patelli, and I. I. Zovko, "The predictive power of zero intelligence in financial markets," *Proceedings of the National Academy of Sciences*, vol. 102, no. 6, pp. 2254–2259, 2005.

[6] D. Ladley and K. R. Schenk-Hoppé, "Do stylised facts of order book markets need strategic behaviour?" *Journal of Economic Dynamics and Control*, vol. 33, no. 4, pp. 817–831, 2009.

[7] D. Ladley, "Zero intelligence in economics and finance," *The Knowledge Engineering Review*, vol. 27, no. 2, pp. 273–286, 2012.

[8] K. Boer, M. Polman, A. Bruin, and U. Kaymak, "An agent-based framework for artificial stock markets," in *In 16th Belgian-Dutch Conference on Artificial Intelligence (BNAIC*, 2004.

[9] M. Cristelli, *Complexity in financial markets: modeling psychological behavior in agent-based models and order book models*. Springer Science & Business Media, 2013.

[10] B. LeBaron, "Chapter 24 agent-based computational finance," in ser. Handbook of Computational Economics, L. Tesfatsion and K. Judd, Eds., vol. 2, Elsevier, 2006, pp. 1187–1233. DOI: https://doi.org/10.1016/S1574-0021(05)02024-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574002105020241.

[11] A. W. Lo, *The adaptive markets hypothesis*. Princeton University Press, 2019.

[12] A. V. Rutkauskas and T. Ramanauskas, "Building an artificial stock market populated by reinforcement-learning agents," *Journal of Business Economics and Management*, vol. 10, no. 4, pp. 329–341, 2009.

[13] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[14] R. S. Sutton, F. Bach, and A. G. Barto, "Ch. 6 temporal difference learning," in *Reinforcement learning: An introduction*. MIT Press Ltd, 2018.

[15] ——, "Ch. 3 finite markov decision processes," in *Reinforcement learning: An introduction*. MIT Press Ltd, 2018.

[16] Towards Data Science. "The complete reinforcement learning dictionary." (), [Online]. Available: https://towardsdatascience.com/the-complete-reinforcement-learning-dictionary-e16230b7d24e.

[17] A. Tucker, A. Gleave, and S. Russell, "Inverse reinforcement learning for video games," *arXiv preprint arXiv:1810.10593*, 2018.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

LEARNING TO TRADE: EXAMINING AGENT BEHAVIOUR IN MARKETS

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Abeillon	Florian
Arnorsson	Sverrir
Yilin	Huang

With my signature I confirm that

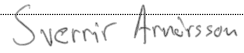
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

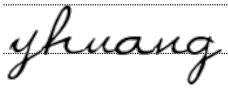
I am aware that the work may be screened electronically for plagiarism.

Place, date

Zürich, December 5th

Signature(s)





For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.