AASMA 21 Project Report

Afonso Gonçalves - 89399
IST - ULisboa
Lisbon, Portugal
afonso.crg@gmail.com

Daniel Seara - 89427
IST - ULisboa
Lisbon, Portugal
daniel.g.seara@tecnico.ulisboa.pt

Mariana Oliveira - 89504
IST - ULisboa
Lisbon, Portugal
mariana.de.oliveira@tecnico.ulisboa.pt

ABSTRACT

The stock market is, without a doubt, an important part in today's society and working with it is supposed to be a private endeavour. However, a recent event has shown that it is possible for a large enough group of people to manipulate it. These people might have worked together, but we cannot forget that the main goal of each one is to make profit for oneself, regardless of others. As such, the problem lies with how cooperation can affect the stock market and how agents can explore this cooperation to maximize their profit. We also try to do the parallelism with the famous Prisoner's Dilemma. A single company has their stocks on the market and a group of agents cooperates in order to increase its price. Moreover, there are random agents adding noise and a single powerful agent with opposing goals. The results have shown us the speculative nature of the stock market and that although cooperation certainly does help, a level of commitment is required by the agents, not only in number but also in willingness to cooperate. Furthermore, this project gave us some insights on how legislation really is necessary for a fair and balanced stock market.

1 INTRODUCTION

The stock market has been present for several years now, and can mean great success both for the individual, who might gain enormous amounts of money, and society in general, with small companies launching an IPO [Fer21] for initial funding and becoming giants (Facebook Inc. Amazon etc). However, great success comes hand in hand with the potential for great loss, be it to an individual who loses his wealth almost entirely, or to whole nations, through crashes with consequences that last several years (see the 2008 US market crash) [Wik21d]. What we know for sure is that it moves large amounts of money, making it an engaging topic to study.

In the beginning of 2021, something disrupted the stock market and shocked many investors: Gamestop (GME) was being short sold, i.e., investors were borrowing stocks and selling them, promising to return them within a deadline and hoping that the price would drop during that time. However, there were more stocks being sold than the existing ones. Some people took advantage of that and collectively bought all of GME's stocks [Fit21]. When the time came for short sellers to buy those stocks back (as part of the short-selling deal), there were no more stocks available, leading to a huge increase of their price. This cooperative behaviour led to an even more interesting scenario: the stock's market value was higher than its real value and it was still rising, but it was sure that at some point it would crash. However, that point was in total control of the cooperative community: as long as nobody sold those stocks, the price would continue to rise. When people start to sell though, the value will start to drop, leading everyone else to sell their stocks as well, fearing the loss of what they had accomplished, making the stock price decrease very fast. This tense equilibrium is somehow analogous to the Prisoner's Dilemma [RCO65] and will be the focus of this project.

As such, this project aims to, firstly, study the **impact of cooperation in the stock market**, specifically on the stock value of a single fictitious company and consequently understand the importance of the legislation that controls the market's dynamic. Secondly, we want to analyze how different populations with different willingness to cooperate behave in a competitive environment and **whether a cooperative strategy may lead to success**.

2 APPROACH

First and foremost, given that our project aims to resemble a stock market (however it is not a trustworthy simulation), there are a few basic rules by which the agents must abide:

(1) An agent cannot buy stock at a bigger price than its current available money;
(2) An agent cannot sell more shares than the ones it owns;
(3) The price of the stock must always be above 0.

We simulate a single company with a fixed amount of shares. The time of the environment is split across several time frames (let us call them "ticks"). In each, every agent can trade its stocks, either by selling or buying them, or do nothing at all. At the end of the tick, the offers are matched using a Double Auction system [Wik21a] with the Average mechanism and the clearing price for the next tick is determined.

More formally, we can characterize this environment as:

- **Accessible**, since the information required to make an informed decision (the last clearing price of each stock of the company) is always available to every agent;
- **Non-deterministic**, because the outcome of an action also depends on what other agents do;
- **Static**, because buy and sell orders are all completed at the end of each epoch, only after every agent has decided its action. That is, there is no change in the environment while the agents decide on their action;
- **Continuous**, since the stock can take any price, meaning there are infinite states;

Lisbon '21, May 28, 2021, Lisbon, PT

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

- **Non-episodic**, since at each run, we allow our cooperative agents to save what they have learned so far and load it when beginning a new run.

2.1 Stock Market

The stock market simulates the environment of our system. At each tick, it performs a Double Auction Algorithm to match buy and sell orders and determine the new stock price. To do so, it uses the buy and sell orders (stored in *buyOrders* and *sellOrders*, resp.) that have been place in that same tick. After each auction, the Stock Market also updates the last clearing price (*stockPrice*). To simplify the system design, the Stock Market notifies the agents that won each auction. It also discards every not-matched order after performing each auction, so agents don't need to keep track of the orders they have currently placed.

2.2 Generic Agent

The way our Market is designed provides a good abstraction for the agents to implement their perceptors and actuators. In fact, agents interact with it by querying the latest stock price (*getStockPrice()*) and by placing sell and buy orders (*placeSellOrder()* and *placeBuyOrder()*, resp.). When placing these orders, each agent specifies how many stocks are associated with the operation and the value given to each. Whenever one of the agent's orders is matched, the agent is notified and updates both *stocks* and *wealth* accordingly. The agents are, however, free to perform no action in a tick.

That said, a generic agent can access the following information:

- *stocks* - the number of stocks it owns;
- *stockPrice* - the last market clearing price, obtained via the perceptor *getStockPrice()*. This information is obtained from the stock market;
- *wealth* - the amount of money the agent has;
- *worth* - the value the agent possesses including the stocks it owns at the most recent stockPrice;

The goal of each agent is to maximize its own worth, despite the other agents' outcome. This metric is indicated by the final worth balance (difference between the final worth and the initial worth). There are 3 types of agents interacting with our stock market: Weak Isolated Agents, Strong Isolated Agents, and Cooperative Agents. Refer to the following subsections for an explanation of each of them.

2.3 Weak Isolated Agents (WIA)

The *WIAs* illustrate the average stock market investors that have weak economic power. These agents act random and independently, introducing noise and unpredictability in the environment. This randomness makes the stock price float around a fixed value, even in the presence of abrupt variations. This allows us to simulate a balanced and realistic stock market. It is essential that these agents represent a significant percentage of the population since it reduces the impact each agent has on the market on its own and creates a more stable environment. The present solution used populations with 66% of *WIAs*.

These agents follow a reactive architecture since they are not the focus of our study. Each *WIA* has a baseline probability to act, *probAct*, which helps the agent decide if it is going to skip its turn or not. This behavior makes sense to us, as ordinary people do not trade every single hour of every single day. That said, when the *WIA* decides to act, it will verify two conditions: whether it has enough wealth to buy new stocks and whether it has stocks to sell. If none are true, then the agent cannot perform any action, even though it decided to act. In case one of this conditions is false, then the agent has no option and performs the other action. Otherwise, it chooses the action with a probability of 50%. Such randomness in the decision making process allows the agent to perform different actions in the same internal state and creates a stable stock market.

Each agent has a greediness factor (*greed*), that indicates at what price the orders will be placed, and a *maxBargain* property, which indicates the maximum amount the agent will bargain with. When placing a buy order, it uses these factors to calculate the price he is willing to pay, normally lower than the current price of the stock. When executing a sell order, the same factors are used to calculate the price at which to sell, most times at a higher price than the current one. The specific calculations can be found in *WeakIsolatedAgent.py, lines [38, 40] and [53, 54]*. They assure the proposed value is different but not too different to the current stock price.

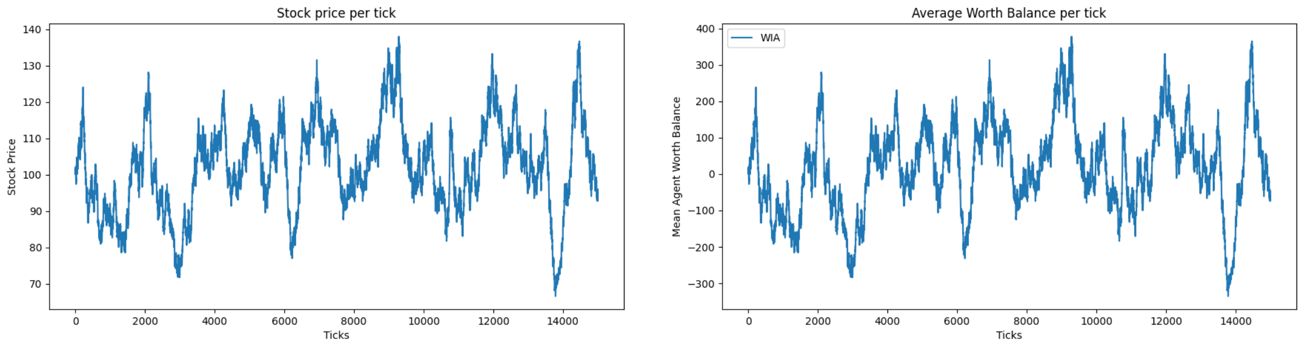
Figure 2a depicts the stock price variation in an environment with *WIAs* only. Despite having a lot of noise, the price stabilizes around 100 monetary units, which shows the stability of the market. Figure 1b shows the average worth balance for these agents at each tick. The average worth however, varies a lot, going from losing 300 monetary units in worth (when the price of the stock drops) to gaining almost 400 (when the price rises). Interestingly, when the simulation ends, the price of the stock is very similar to the initial one, as well as the average worth, showing that after all is completed, the agents have not really gained or lost anything. This is due to the random nature of the *WIAs* and the work done to balance the aforementioned probabilities. We consider this a good baseline to start introducing other types of more intelligent agents.

2.4 Strong Isolated Agents (SIA)

The environment has a single *SIA* that represents the Hedge Funds in the described scenario, and we can say it is our main adversary. As such, it has a strong economic power and can, by himself, influence the market. This agent is short-selling, thus its worth is maximized by lowering the stock price. The mechanism of short selling can force this agent to have negative balance.

Since it is not the main objective of our study, this agent also has a reactive architecture, albeit a more "intelligent" one. It also has a different action set: instead of selling and buying stocks, a *SIA* can short-sell and re-buy them. The first operation allows it to sell stocks it do not possess, creating a debt equal to that number of stocks. The *SIA* must pay their debts within 1000 ticks after its contraction. When a *SIA* re-buys a stock, then it is deducted from the oldest debt it contracted.

Every tick, the agent stores the current *stockPrice* and uses this information to take its future decisions. It then considers every debt it has at that moment and chooses if it wants to re-buy the stocks in that tick or not. This agent will re-buy a stock if it has found a valley in its *stockPrice* history and predicts that the price will not



(a)

(b)

Figure 1: Run with WIAs only

drop in the near future. It does so by identifying a stock price in the last 5 ticks that is lower than the last 75 recorded stock prices. If the stock prices rise above a defined threshold after that minimum, then the *SIA* has found the valley. If the *SIA* passes the deadline of a debt, then it will be forced to re-buy those stocks at any price.

After handling its debts, the *SIA* decides if it will short-sell in that tick. The agent will short-sell if the following conditions are met:

(1) If it has no current debts;
(2) If it has sufficient information about the stock price history;
(3) If the debt deadline that results from that short sell would happen within the simulation;
(4) If it randomly decides to act;
(5) If it found a peak in the stock price history. A peak is analogous to a valley, as the agent looks for a price drop after a recent maximum.

When a *SIA* short sells, it opens a sell position for 40% of the number of stocks currently available.

Figure 2 shows the impact of having a *SIA* in the environment. We can see that the stock price graph is very similar to the one with only *WIAs*, but the most interesting part is in the graph to the right hand side. Due to the mechanisms implemented in this agent, we can see that its worth kept rising throughout the simulation, even when the stock price was lowering, and the weak agents were losing worth. This is exactly the expected result. A strong agent, with a lot of economic power and more intelligent than the weak agents, is able to take advantage of the market and short sell stock in order to make money. In fact, by overlapping both graphs it is noticeable that when the price goes down, the *WIAs* lose worth, but the *SIAs* manage to profit considerably. With this environment, we are now in the best position to analyse the impacts cooperative agents have.

We could have decided to implement more than one *SIA*, since there are more than one Hedge Funds in real life. However, given our model, one of these agents is already strong enough and achieves our purpose. As such, we decided to follow the Occam's razor principle and not add more, avoiding unnecessary complexity.

This mechanism provides us with the best balance between an agent that can influence the market, and a relatively simple architecture that was rapidly implemented and did not need to learn during our simulation.

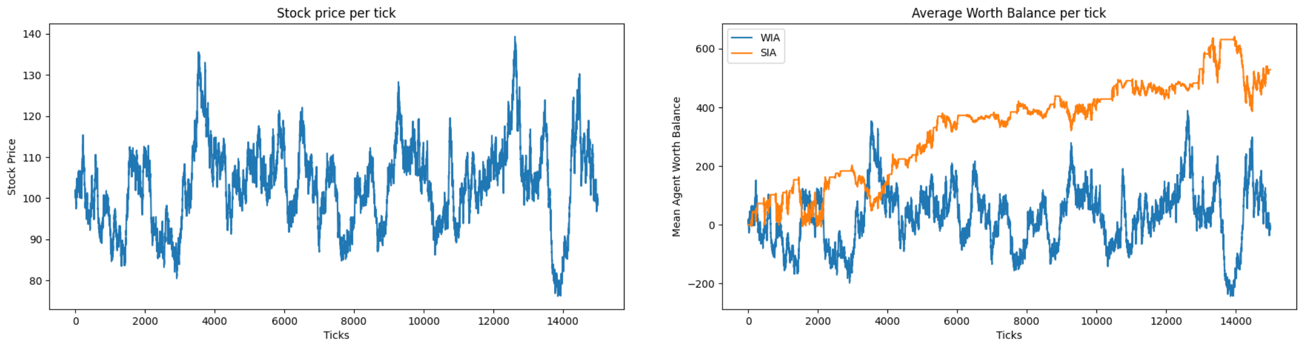
2.5 Cooperative Agents (CA)

The *CAs* are the main focus of this project: they have weak economic power, much like the *WIAs*, but the combined economic power of every *CA* is enough to affect the stock price. Similarly to the other types of agents, they have a probability to act, *probAct* and as the *SIA* they keep an history of stock prices, which is used for learning purposes. In addition, these agents also keep a list of other *CAs*, *peers*.

The *CAs* can cooperate with each other by broadcasting their action to their *peers*, as an invitation for them to take the same decision. This cooperation is instantaneous, i.e., at any tick each agent can choose to cooperate regardless of its previous decisions. Note that we do not allow any sort of malicious communication, meaning if *agent1* receives information according to which *agent2* will execute action *a*, then it can be sure action *a* will be executed by *agent2*. In other words, agents cannot lie. An agent *defects* when it stops cooperating. The willingness for an agent to cooperate is translated in a value, $probCoop \in [0, 1]$. Such attribute can be changed between runs of our system.

Since we have a sequential mechanism for executing an action (a simple *for* loop that calls the *act()* method of each agent), we understand that the first agents of the tick are at a disadvantage, having to make a decision with little information. We mitigate this problem by randomly changing the order in which these agents act at every tick.

A cooperative agent can decide on one of four possible actions: *BUY*, *SELL*, *SKIP* and *HOLD*. The first and second correspond to the generic agent primitives with the same name. The *SKIP* action can be seen has not doing anything. The last one represents the moment when the agent has stocks but is deciding not to sell them, even though it could have made a profit. This represents exactly what happened in the scenario that motivated the development of this project.



(a) **(b)**

Figure 2: Run with WIAs and a SIA

We started by implementing a BDI [Wik21b] architecture. At each tick, each agent would update its beliefs and generate its desires based on them. Each desire would be associated with a strength level that represented how much the agent wanted to perform each action. This strength would depend on other agents' actions and on the stock price history. Finally, the agent would randomly choose one of those actions, giving more chances to the ones associated with a higher strength. Before executing the action, the agent would randomly decide if it was going to cooperate or not, having a probability of 50% for each option.

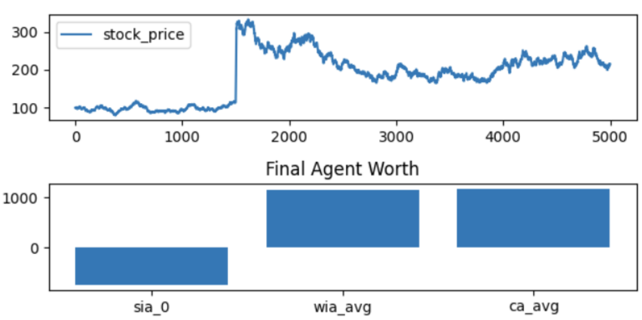


Figure 3: Stock price variation and average final worth per type of agent, with the BDI architecture

Figure 3 depicts the satisfactory results given by this implementation: Contrarily to the *CAs*, the *SIA* was not able to profit. We can also note a sudden increase in the stock price, that resulted from the *SIA* having to re-buy its stocks after the deadline.

Despite these satisfactory results, we wanted these agents to be able to learn and improve their decision making by themselves. As such we implemented the Q-learning algorithm [Wik21c]. We started by gathering the factors that could influence each agent's decisions and concluded that they would be the following:

(1) Current stock price trend;
(2) Other cooperative agents' actions (whether they are buying, selling, skipping or holding);
(3) If the agent has stocks to sell;

Percentage	Classification
[0, 20[	**l**ow
[20, 50[	**m**edium
[50, 100]	**h**igh

Table 1: Peer action percentage classification

(4) If it has enough balance to buy more stocks.

One big challenge of this approach was to summarize all the above information in a finite number of states. We had to discretize it, as explained below:

Firstly, the agent checks the stock price variation in the last 10 ticks. If the price is dropping, it records a *D*, otherwise it records a *U*. Secondly, it analyzes what its peers are doing, by calculating the percentage of peers for each action that it received. These values are discretized in 3 classes, as shown in Table 1. Finally the agent checks 3) and 4), assigning *T* if these conditions are true and *F* if they are false. The computed state results from the combination of these assertions.

For example, state **UBlHlShKlTT** tells that:

- U - the stock price is rising;
- Bl - there is a low number of peers buying;
- Hl - there is a low number of peers holding;
- Sh - there is a high number of peers selling;
- Kl - there is a low number of peers skipping;
- T - the agent has stocks to sell;
- T - the agent has enough money to buy stocks;

The reward is given by the worth tick balance of that round ($reward = worth_t - worth_{t-1}$). We set our α to 0.01 and our γ to 0.99.

It is then just a case of, at each tick, given the action performed in the last state, calculate the reward and update the Q-value of the previous state. After this, the agent gets its current state, selects an action using the ϵ-greedy strategy (ϵ is set to 0.05) and waits for the next tick.

One thing we also implemented is the ability for the agents to learn from previous runs of the simulation and from the experience of other, more successful agents. As such, at the end of each run, our system saves the Q-values for the agent that has the highest worth and, when another run is started, all agents read from these values and use it as their initial Q-matrix.

3 EMPIRICAL EVALUATION

We run our model in a normal laptop, simulating 15000 ticks. The stock price starts at 100 units and each weak agent, including cooperative ones, has 10 stocks and a 1000 monetary units (henceforth using the € symbol). The strong agent is given 200 000€ and no stocks. Before running these tests, we ran the simulation 2000 times.

Firstly we wanted to study the impact of having cooperative agents in the stock market environment and ran our default experiment, having 33% of cooperative agents, each with a 50% chance of cooperating with its peers. The results for this experiment are available in Figure 4.

We can compare these results with the ones obtained in Figure 2. The most noticeable result is that the stock price revolved around 200€, which is two times the value in the other test. This shows the strong impact coordinated agents can have in the stock market, even though being a minority of the total population. We can also note that even though the *SIA* was able to make a similar worth profit, the *CAs* doubled its final worth balance, once again suggesting the power of cooperation. We also observed that despite having a high worth balance, these agents had a negative wealth balance, meaning that in the end of the run, they lost money and could not convert their stock value into monetary value. This negative wealth balance accentuates the value the *CAs* were able to gain in stocks and is explained by the fact that these agents used the worth balance as a reward function, instead of their wealth.

We then tried to understand the impact of having different percentages of cooperative agents in this environment and ran 2 more tests, having 70% and 80% of cooperative agents in the population. We kept the *probCoop* at 50% as to ensure that any change in the results stems from the size of the population and not its cooperative nature. These results are shown in Figure 5.

Graphs 4a, 5a and 5b show an interesting result. Almost immediately, there is a big jump in the price of the stock from 100€ to 240€ in the first case and 600€ or even 800€ in the others. This happens in the exact moment after the strong agent has decided to initiate a short sell position. Since it has sold a lot of stocks, these agents are buying and holding them, preventing the price from going down and the strong agent from making a profit. Such is supported by graphs 4b, 5c and 5d. The average worth of the cooperative agents shoots up and the *SIA*'s take a dip. After that, we enter a phase where the cooperative agents are in control of the market and each one can now decide to keep holding or selling to make a profit and to cooperate or not with the rest of the agents. As it is to be expected, with more cooperative agents, the average worth balance of the agents is higher, going from a peak of about 2000€ with 33% to a peak of almost 14000€ with 80%.

For graphs 4b and 5d, we can actually see the cooperative agents are not able to maintain the advantage over the *SIA* and, as we can see by the orange line, it starts clawing back up and increasing its worth. Graph 5c tells a different story. With these values for cooperation, the agents are capable of punishing the *SIA* and not let it increase its worth.

Let us look more carefully into this, since it is our main focus. From Figure 4 there seems to be some sort of Prisoner's Dilemma in action. When most of the agents decide to cooperate and, for example, hold, their worth goes up. However, when all decide to sell, the market is flooded with stocks and the price goes down. By opposition, if they do not cooperate they can make money if they sell and others do not, but they cannot make the price go even higher for higher potential profits.

We now focus on how the simulation ends in greater detail. Once again we will examine the impact of increasing the size of the cooperative population (with $probCoop$ = 50%) and then of increasing the willingness to cooperate of a population of fixed size (33%).

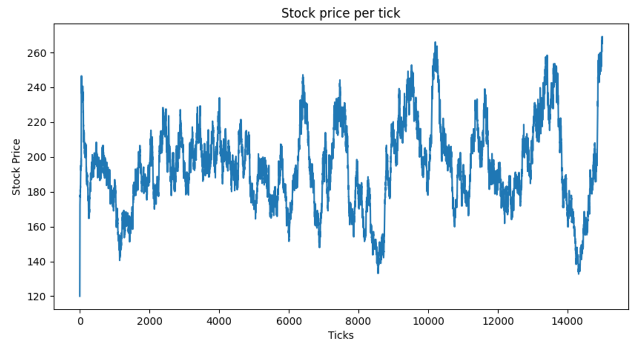
Regarding the former point, according to the graphics of Figures 4 and 5, independently of the size of the population, the *CAs* always increase their worth although such increase is more significant the higher the percentage of cooperative agents. However, the final worth balance of the *SIA* behaves differently: both with 33% and 80% it increases, but with 70% of *CAs* it becomes negative.

The *WIAs* worth balance behaves much like that of the *CAs*, but its change is less expressive. Let us now on graphs 5e and 5f, the graphs related to the wealth balance. The *CAs* always lose money, which is expected. Given they are trained to increase their worth, not their wealth, they end the simulation with many stocks. They would only translate to money if they decided to sell them. Nonetheless, note that despite losing money, such loss is inversely proportional to the size of the cooperative population. Once again, the *SIA* only loses money when 70% of the weak population is cooperative. In the other two configurations it makes money. As for the *WIAs*, the increase in their wealth balance is proportional to the percentage of cooperative agents.

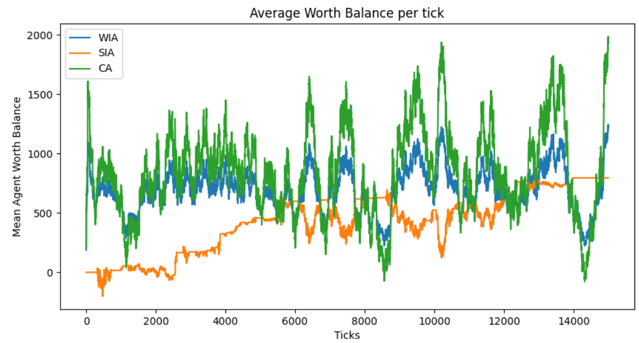
We also wanted to understand the immediate impacts of cooperation to understand if agents had any incentives to. We managed to gather the tick worth balances of every cooperative agent for every tick and split them in two categories, according to whether they resulted from cooperating or not.

In Figure 6, we can see that there is no significant immediate advantage in cooperating vs not-cooperating, regardless of the percentage of cooperative agents. However, at first glance, this graph can be misleading: on average there is no change in the agent's worth, but there are a lot of outliers that can make up to 1500€ in the best case scenario. What this means, we hypothesize, is that cooperation is a long term investment and a commitment. The agent cooperates and its worth improves, but only after several ticks.

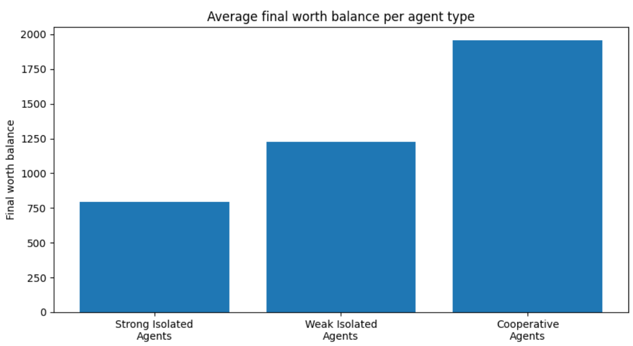
Finally, we wanted to see the effects of having agents with different probabilities to cooperate. We ran this simulation with 33% of cooperative agents, each having 75% of probability to cooperate. Worth wise, all types of agents profit with a greater cooperation among the *CAs*. However, such improvement is much more noticeable for the *SIA*, which experiences an increase in worth balance of about 300%. In terms of wealth, the increase in *probCoop* does not cause the *CAs* final wealth balance to chance significantly. Conversely, the *SIA* not only continues making money, but increases its payoff by about 300% as well.



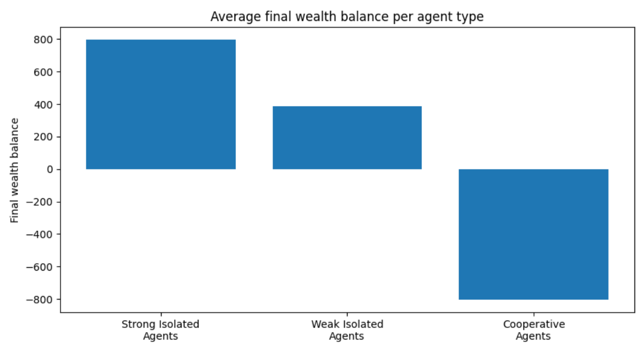
(a) Stock price per tick



(b) Average Worth balance per tick



(c) Final Average Worth balance per agent type



(d) Final Average Wealth balance per agent type

Figure 4: Execution with 33% Coop Agents; 50% prob. to cooperate

4 CONCLUSION

All the aforementioned results allowed us to reach a few conclusions regarding our scenario and our objectives for this project. In fact, cooperation can indeed manipulate the market and help the agents participating in it and potentially make a lot of money. However, they need to be 50 to 75% committed to cooperating and not be tempted to defect in order to reap the best profits. That said, it becomes clear how important legislation is in preventing and regulating such cooperative movements. With respect to the average investor, our results support the hypothesis that their worth / wealth increases proportionally to the level of cooperation in the environment, regardless of such cooperation being the result of a larger cooperative population, or of a higher willingness to do so by the *CAs*.

Our system also demonstrates how speculative the stock market is. Despite having a considerable increase in worth, on average, our agents lost wealth. It is important to keep in mind that the worth is only useful if the agents are able to convert it to wealth, i.e., sell their stocks.

We can also relate this situation to the Prisoner's Dilemma: if every cooperative agent is holding their stocks and not selling them, the price will continue to rise. This is an incentive for agents to deviate and start selling their stocks, so they can increase their wealth. However, if every agent does this, the stock price will drop and every cooperative agent looses worth and wealth. We would be excited to further explore this situation by trying to create agents that aim to maximize their wealth instead of their worth, since this would better represent this dilemma.

Finally, the work in this project has, indubitably, helped us learn a lot about how to design and implement a multi-agent system. Moreover, it allowed us to work on a topic we all enjoy, the stock market, and its implications in today's world. Although our simulation is, by no means accurate, it is enough to give us an intuition into how it works and the profit that can be made from it.

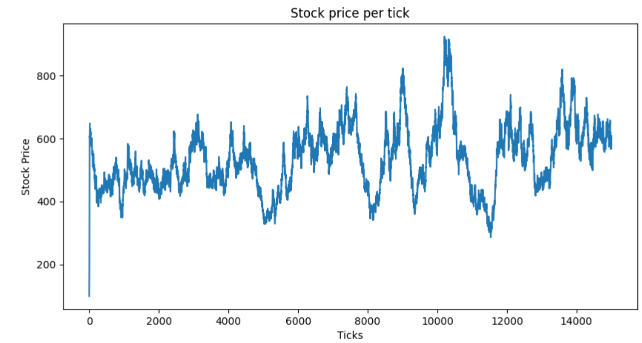
In future iterations, we would like to let our simulation run a lot more times, in order to get a more stable Q-matrix, extracting the policy from it and build the decision tree associated with our agents decisions, so as to better understand the rational behind them and also see if there was any type of error that needed to be fixed.

All in all, we really liked this project and the work we were able to produce.

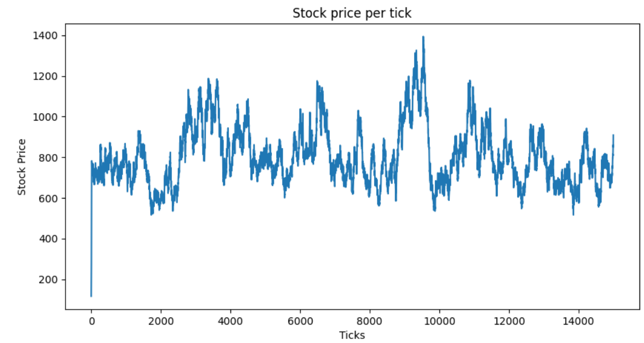
REFERENCES

[Fer21] Jason Fernando. Initial public offering (ipo), 2021. https://www.investopedia.com/terms/i/ipo.asp.

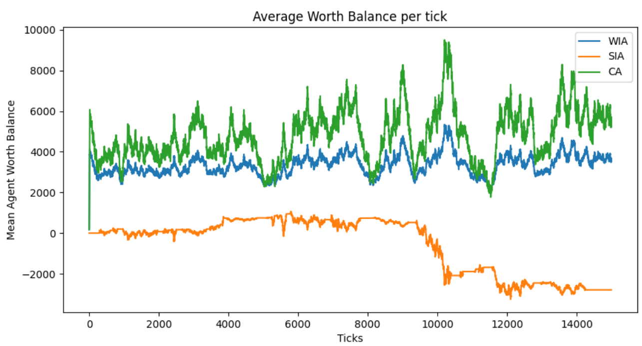
[Fit21] Dan Fitzpatrick. What really happened with gamestop $gme ?, 2021. https://stockmarketmentor.com/2021/01/what-really-happened-with-gamestop-gme/.



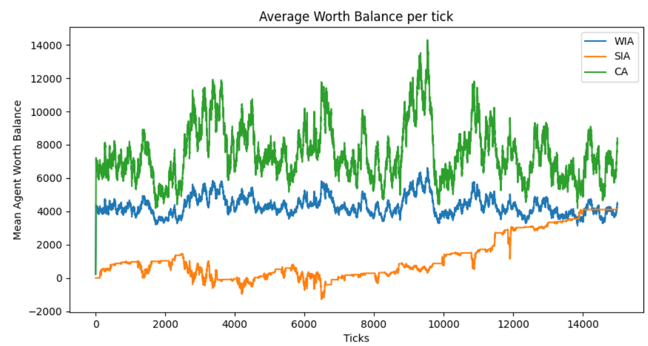
(a) **Stock price 70%-50%**



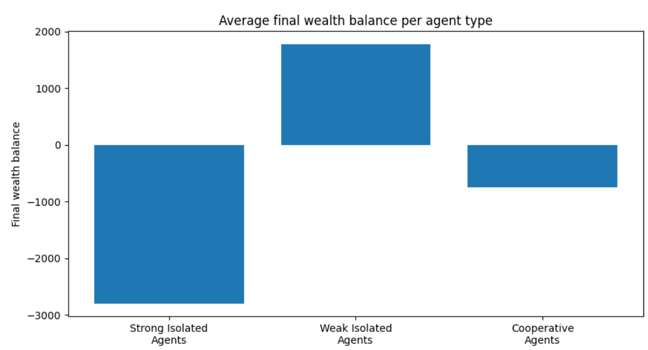
(b) **Stock price 80%-50%**



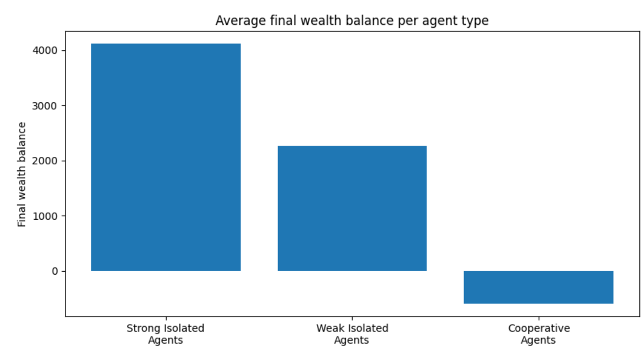
(c) **Average Worth balance 70%-50%**



(d) **Average Worth balance 80%-50%**



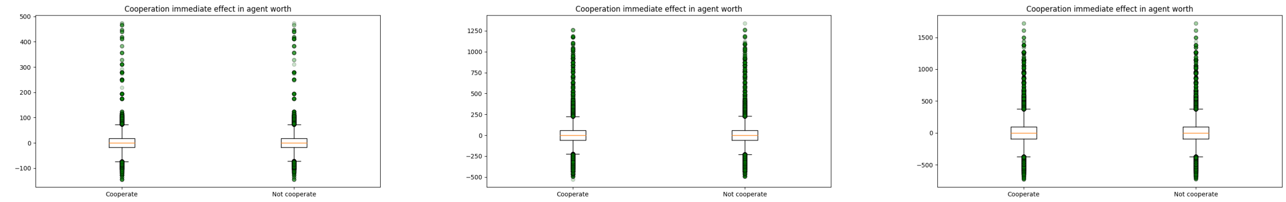
(e) **Final Wealth balance 70% - 50%**



(f) **Final Wealth balance 80% - 50%**

Figure 5: Stock Price, Worth and Wealth balances for 70% and 80% of Cooperative Agents

[RCO65] Anatol Rapoport, Albert M Chammah, and Carol J Orwant. *Prisoner's dilemma: A study in conflict and cooperation*, volume 165. University of Michigan press, 1965.
[Wik21a] Wikipedia. Double auction, 2021. https://en.wikipedia.org/wiki/Double_auction.
[Wik21b] Wikipedia. Double auction, 2021. https://en.wikipedia.org/wiki/Belief%E2%80%93desire%E2%80%93intention_software_model.
[Wik21c] Wikipedia. Double auction, 2021. https://en.wikipedia.org/wiki/Q-learning.
[Wik21d] Wikipedia. Financial crisis of 2007-2008, 2021. https://en.wikipedia.org/wiki/Financial_crisis_of_2007%E2%80%932008.

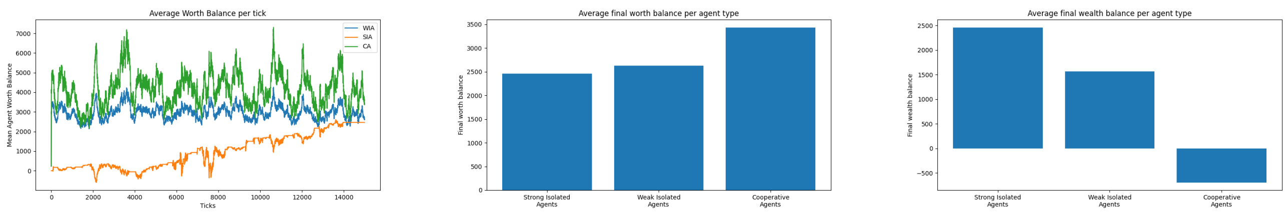


(a) **33% Cooperative Agents - 50% prob. cooperation**

(b) **70% Cooperative Agents - 50% prob. cooperation**

(c) **70% Cooperative Agents - 50% prob. cooperation**

Figure 6: Immediate effect of cooperation



(a) **Average Worth balance per tick 33% - 75%**

(b) **Final Worth balance 33% - 75%**

(c) **Final Wealth balance 33% - 75%**

Figure 7: More communicative agents (75%)