ABSTRACT

In sociology, the social exchange theory argues that any social phenomena, e.g. war, marriage and friendship, can be decomposed into a series of interactions, where one property or another is being exchanged. In this project, the social exchange theory will be applied in the interactions between intelligent agents in a simulated game environment. In the game, the agents will use their skills to compete for gold. These agents are given the options to mentor, steal from, help and befriend another agent, and this decision will be influenced by the agent's personality and skill level. This project has demonstrated that social phenomena can form through the simple application of the social exchange theory without the need for coercive methods. Specifically, stable power hierarchies and social structures were the only phenomena detected and in most of the places, where one of them could be detected the other could as well. The critics of the theory often point to collective action as behaviour that cannot be reproduced through the social exchange theory alone, and such behaviour couldn't be detected in this project.

TABLE OF CONTENTS

1 INTRODUCTION

All people are bound to the effects of social phenomena by virtue of existing; common societal structures such as families and friendship groups, as well as more complex ones, such as governments, are all examples of social phenomena that have a large impact on our lives. Due to the growing advancements in computation, researchers now have more techniques for studying it; in fact, the emerging field of agent-based social simulation (ABSS) combines social science, agent-based computation and computer simulation techniques to gain a deeper understanding of social phenomena (Lie et al, 2008).

The social exchange theory is a pivotal theory within sociology that seeks to explain the cause of this phenomena. It works on the assumption that human beings are rational and so will act in ways to maximise their gain by weighing up the costs and benefits of certain actions. In more detail, it sees society as a series of different interactions between individuals, and each interaction is maintained so long as the rewards outweigh the punishments. The major criticisms of this theory lie in its assumption that people always act rationally, the question of why do people help others at the expense of themselves (altruism) and questions on how it can explain the formation of social structures and norms.

This project is an investigation into the problem of identifying social phenomena, particularly collective action, social structures formation and power distributions, through the application of the social exchange theory on agents interacting in a social environment. This project can be decomposed into simulating the social exchange theory and identifying emergent properties from this simulation. The first problem involves emulating people (via certain characteristics) as intelligent agents, providing a framework for agents interactions, and developing a social environment for these agents. The second problem consists of developing detection software to identify social phenomena and designing an experiment to efficiently search for configurations of agents and simulation settings, where such properties emerge. The main goals of this project are to understand how complicated social behaviour evolves through the application of the Social Exchange Theory and to identify insights on how it can be used to affirm or respond to some of the aforementioned critiques.

The next chapter examines the research on the Social Exchange Theory and the approaches others have taken in simulating interactions. Chapter 3 explains and justifies the approach taken by this project. The following chapters 3 and 4 explain the designs, implementation and tests undertaken for this approach. The results have shown the emergence of social phenomena, amongst the agents within the simulation, in the form of stable power distributions, such as repressive and democratic governments, and social structures. These results are explained, in more detail, in chapters 5 and 6. In the final chapters, the implementation and results are evaluated to determine how successful this project has been in achieving the above goals.

2 REVIEW OF LITERATURE

The purpose of this literature review is to explore Social Exchange Theory in terms of its critiques, counter-responses and applications, to understand how it can simulated. Also, other approaches to agent based social simulation or modelling will be evaluated.

2.1 Social Exchange Theory

One of the larger critiques of the Social Exchange Theory lies in the problem of collective action. Specifically, how can individuals, who wish to maximise personal gain, act in a way that benefits other people more so than themselves? Unions are commonly introduced within the literature to exemplify this problem; for example, Scott 2000, argues that since unions work to benefit all members of the profession, regardless of union membership, rational actors have no need of paying union fees or dedicating time in being active participants, and yet, many people still decide to join such organisations. The idea of selective incentives, which are rewards made available to people for contributing to a collective good, explains how collective behaviour can manifest between self-interested actors. In the case of unions, the selective incentives may be beneficial legal and career advice as well as the opportunity to build useful connections.

Since the Social Exchange Theory posits that all social phenomena are reducible to individual action, some sociologists question how it can explain the formation of large social structures as well as the complex behaviour exhibited within them. Social exchange theorists respond to such critiques by presenting the notion that large social structures (and so the behaviour exhibited by them) are generated through the formation of many different individual exchange relations (Cook, Emerson, Gillmore and Yamagishi, 1983), or in other words, they are unintended consequences of individual action (Scott, 2020). However, there are critics who doubt this 'micro-to-macro' explanation of social structure; for example, Zafirovski considers it to be reductionist because it disregards self-emerging properties of macrostructures that cannot be decomposed into individual exchanges.

A team of researchers in Norway studied the Social Exchange Theory as an explanation of organisational citizenship behaviour (OCB) amongst teachers (Elstad et al, 2011). OCB is a form of collective action defined as voluntary behaviours and actions of members that exceed their expected workload. In this study, 234 secondary school teachers were surveyed and the results showed that trust within principal and teacher interactions is important in fostering collective behaviour. This study shows how collective action can be understood in a real-world context through the lens of the Social Exchange Theory, which is a useful perspective to refer to throughout this project.

2.2 Agent based social simulation

The abilities to both implement and then evaluate a new social theory have made social simulation studies a useful tool in the field of sociology; with some even seeing it as the natural bridge 'between empirical research and theoretical work' (Lie et al, 2008). The next two approaches are based on organisational modelling, which is an especially popular field within ABSS that is closely related to identifying forming social structures amongst simulated agents.

Grossi, Dignum, Dastani and Royakkers (2005) are researchers who, using multi-modal propositional logic, proposed a framework to formally model organisational structures in multi-agent systems (MAS). Their work highlighted how the refinement of social norms is essential in constructing organisational structures within MAS. They defined these organisational structures as structures that emerge to achieve specific objectives, which are distributed within the organisation as agent roles. Social norms are placed on certain roles to restrict the types of actions that can be taken during interactions. Both of our research involves finding organisational structures emerging from a multi-agent system. However, their research uses social norms to coerce this behaviour, whereas this project investigates whether such behaviour can emerge naturally.

Another example of using social simulation to study emerging social phenomena is the work done by Latané, Nowak and Liu, 1994. Specifically, their work involved distributing attitudes, before the start of a discussion, into two groups of those who have the majority viewpoint and those in the minority. They simulated how these attitudes can change throughout a discussion by applying social influence laws from sociological literature. This simulation environment consisted of a grid arrangement of agents, who could see and hear their closer neighbours more clearly than their distant ones. In their results, they were able to see self-organisational tendencies, where agents with a similar attitude within the simulation grouped together, as emergent social phenomena from analysing group properties. Their research shares many similarities with my project seeing as both of them involve simulating social interactions and investigating emergent social phenomena. However, their research uses group processes to change the whole distribution of attitude so the agents cannot be seen as rational actors, which act to maximise their own benefit, they instead represent attitude states at specific positions; therefore, while it can be said that their 'agents' interactions involve the exchange of the social token 'attitude', this exchange occurs at a group (or top-down) level instead of at the individual (or bottom-up) level that the social exchange theory operates on.

Although the previous examples are unhelpful in the problem of developing personalised agents, there is a substantial amount of literature on these types of agents. Personalised agents are 'intelligent' agents designed to emulate how a person would act in a given environment; they typically are assigned specific goals, and behaviours, as well as, human characteristics, such as personality and an array of physical traits. Of course, this begs the question of how does one choose such characteristics for their agents?

According to many researchers, personality and physical attractiveness are accurate and distinct predictors for status in face-to-face groups (Anderson et al., 2001); one researcher even found that personality is one of the most accurate, even more so than cognitive ability, predictors of mortality, divorce and occupational attainment, which have a large predictive effect on socioeconomic status (Roberts et al., 2007). Since status is an essential component of social interaction, personality is an important human characteristic needed in personalised agents. Out of the many representations of personality, the Big Five Factor model (McCrae and John, 1992) is the most universally accepted model due to its prevalence and strong biological basis since each factor can be mapped to certain brain structures (DeYoung et al., 2010).

Li et al., 2007 created a personalised (FFM model) learning companion agent who could demonstrate certain behaviours based on its mood, personality and emotional state. This was improved further by Zhen Liu, 2008, who created a personality model of virtual characters. This model also gave characters a personality and a situational emotional state but it extended the previous approach by assigning physical and social (e.g. occupation and race) traits as well as specific motivations and behaviours, which a character can use to decide what actions to take. A team of researchers developed this model even further by introducing anxiety and applying this model to agents in a goalkeeping football simulation to accurately show that highly neurotic goalkeepers tend to concede more goals (Karimi and Kangavari, 2012). The challenge in implementing these models is due to the large number of states involved, reducing the feasibility of simulating adequate learning behaviour.

This project is reliant on a suitable social environment to house these intelligent agents during the simulation. Within the field of ABSS, there tend to be two types of simulation environments used; generalised ones, where society or scenarios within one are simulated as a whole (although there is usually a focus on specific parts as with any abstraction), or environments based on game theory, where a game is constructed with different assumptions to contextualise a specific social scenario.

Whitworth and Sylla, 2012 developed a generalised social environment model, which contains an outer social environment, named the world, and many various inner ones that function due to a combination of competition and cooperation. The strengths of this model are in its high generality as it can be applied to any group of people and its high validity due to it being based on extensive sociological research. However, this model is limited by the lack of a formalised mathematical representation, introducing ambiguity by leaving some parts up to interpretation. Furthermore, with general models, there's a greater risk of encountering the 'trap of verisimilitude', which is defined as 'the temptation to add plausible detail to the model simply because it can be added' (Doran and Gilbert, 1994).

The games that feature in ABSS problems are typically ones that can capture social dilemmas, which occur when rational actors are forced to choose between actions that are good for the individual but bad for the collective or vice versa; Prisoners dilemma, Tragedy of the commons and the Dictator game are all classic game theory examples.

Snellman et al, 2019, studied the formation of social structures by developing an agent-based model of people playing a hybrid of the ultimatum and dictator game. In the dictator game, one player, the dictator, will divide a fixed reward between the other players. The ultimatum game is the same as the dictator's game with the added condition that if the other players do not accept the amount given no player will receive any reward. Their research showed that agents playing these games were able to consistently form social structures. Although both our research is focused on finding social structures emerging from agents playing a game, their research has a low number of different types of observable social interactions, therefore their observable social phenomena are limited mostly to social hierarchies and organisation formation.

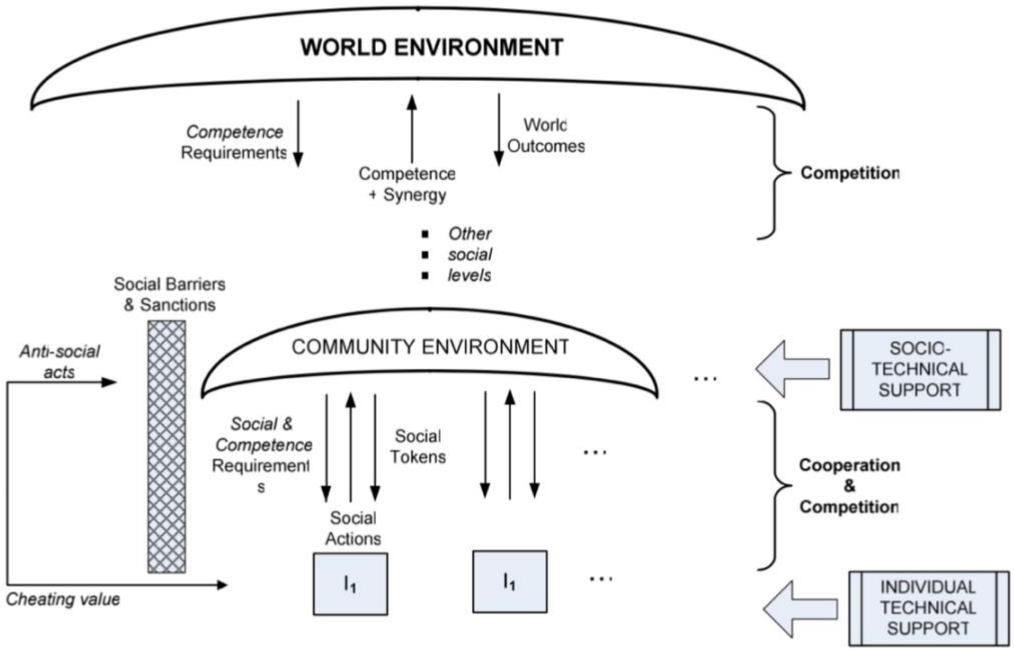
In summary, while there are many examples of projects that focus on finding emerging social phenomena from agent-based simulation, there are few that are trying to achieve this from the bottom up without adding any social norms as constraints and even fewer that are looking for more phenomena besides organisational behaviour.

3 MAIN APPROACH

3.1 Simulating social exchange theory

A simulation environment is the first component required for this project. A trade-off can be identified from the literature between the generalised and game theoretic approaches of selecting a social environment. Generalised models offer a greater potential for emergent social phenomena whereas game-theoretic approaches offer specificity and formalisation. A hybrid between these two methods was used to capitalise on this trade-off.

Figure 1. Social Environment Model by Whitworth, B. and Sylla, C. (2012)



The key information captured by this model is:

- Nested social units: Social units are environments within other environments
- Social tokens: A transferrable property that determines a person's place in a particular social environment e.g. money, grades, and followers.
- Competition is present and there exists competencies within an environment that allow for some to get ahead of the competition.
- The environment punish individuals when they commit anti-social acts.

The chosen simulation environment was a game constructed to contextualise the above model. The game consists of players competing for gold in an environment containing a mineable gold resource. The game can be played by at least 2 players and consists of multiple rounds. Each round starts with an interaction window, where agents are allowed to interact with each other, and ends with a mining window, where the environment distributes gold to the players based on their mining competency. The goal for the players of the game is to generate as much gold as possible. The interactions are bilateral exchanges between two players that take the following forms: Friendship, where each 'friend' (player) will be awarded at the end of the mining window with an additional 10% of all of the earnings of the friend in the current round; Mentorship, where a 'mentee' with a lower competency skill than the 'mentee' can increase this skill by 15% of skill difference in exchange for 20% of all of its earnings in the current round; Help, where a helper will donate 5% of the money earnt by the beneficiary in the last round to the beneficiary; Theft, where a thief can steal 30% of the money earnt by the victim in the last round, but will not be able to participate in the next round if caught. See appendix A for detailed rules of the game.

Justification for this game

The purpose of the game is to facilitate social exchange with gold as the social token. The necessary properties of the generalised social environment model have been incorporated into this game; the players are competing for gold and have a mining competency, anti-social acts are possible with theft interactions and there are sanctions in the form of imprisonment. Adding help interactions has introduced the potential for social dilemmas since helping others, in this case, is a purely altruistic action. Finally, by including friendship interaction, the game can be analysed in terms of social synergy, which is defined as the difference between what individuals produce as a social unit compared to individually (Whitworth and Sylla, 2012).

3.2 Emulating people as agents

The next problem to solve is the problem of developing agents, designed to emulate people, to play this game. The agents have been assigned personalities since this attribute has been shown to strongly influence social interaction. The personality of agents will be represented using the HEXACO model, which is an extended version of the FFM. This model has been identified as a particularly good measure of social and ethical behaviour (Ashton and Lee 2008), and so it will be particularly useful in creating assumptions on how darker personalities influence the types of interactions that these agents choose.

Figure 2. The dimensions and facets of the HEXACO model

Scale	Reliability	Scale	Reliability
Honesty-Humility	.92	Agreeableness	.89
Sincerity	.79	Forgiveness	.88
Fairness	.85	Gentleness	.77
Greed Avoidance	.87	Flexibility	.75
Modesty	.83	Patience	.80
Emotionality	.90	Conscientiousness	.89
Fearfulness	.84	Organization	.85
Anxiety	.84	Diligence	.79
Dependence	.85	Perfectionism	.79
Sentimentality	.81	Prudence	.78
Extraversion	.92	Openness to Experience	.90
Expressiveness	.84	Aesthetic Appreciation	.86
Social Boldness	.86	Inquisitiveness	.81
Sociability	.79	Creativity	.79
Liveliness	.85	Unconventionality	.80

Also, agents will be assigned a mining skill representing how much gold will be distributed by the environment to an agent, and an appraisal skill that represents how accurate an agent is at predicting the personality and competency of the other agents. Having accurate information on other players simplifies the process of winning because it allows for a player to make better strategies.

These ‘intelligent agents’ have been given a learning mechanism, ‘awareness’ on other agents, and memory on observed interactions. The purpose of the learning mechanism is for agents to learn the best strategies to maximise gold earnings given their personality and competency. The agents will be continuously aware of the total amount of gold that all of the other agents have and it will memorise all of the mentorship, friendship and help interactions that occur between the other agents. In regards to theft interactions, agents will only be notified of their occurrence if they are either involved in the interaction or if the other agent has been caught.

As is common in game theory and especially in ABSS, assumptions have been made on how agents' characteristics will influence the game.

Figure 3. Key Assumptions made on how agents characteristics will influence the game

#	Area	Assumptions
4	Friendship	Increases as (and in order of importance): Number of theft interactions between them decreases Number of friendship interactions between them increases Number of help interactions between them increases Similarity in personality increases Number of mentorship interactions between them increases Number of help interactions increases Number of caught theft interactions decreases The higher the level of forgiveness (A1) an agent has, the lower the effect of theft interactions have in decreasing friendship
5	Stealing aversion	Agents high in agreeableness (A) and honesty-humility (H) are less likely to steal Agents are less likely to steal from **friends**
7	Help	Agents are less likely to help wealthier agents. Willingness to help is proportional to their level of **stealing aversion**.
9	Theft	Cost: Increases with **risk aversion** and imprisonment likelihood Potential gain: Increases as the positive difference between the victim and this agents wealth increases Agents are more likely to steal when: Their **stealing aversion** is lower Potential gain outweighs perceived cost
10	Interaction	Agents high in extraversion and openness to experience are more likely to interact with other people
11	Teachable	Agents are more teachable when: High in curiosity (O2) High In self-awareness (O3) High in conscientiousness (C) Low in anxiety (E2)
12	Teaching	Agents are better mentors when: High in flexibility (A3) High in patience (A4) High in fairness (H2) High in sincerity (H1) Highly **teachable**

13	Willingness to be mentored	Need: Increases as the skill-difference between the agent and its mentor increases. Cost: Increases as potential mentorship cost increases Agents are more likely to be mentored if … Agent is highly **teachable** Mentor is a good at **teaching** Mentor is their **friend** The need for a mentor outweighs the cost of mentorship
14	Willingness to mentor	Agents are more likely to mentor if … Agent is highly **teachable** Mentee is their **friend**

See Appendix F for full list of assumptions

3.3 Detecting emergent social phenomena

The final problem solved is the identification of social phenomena from the properties that emerge when intelligent agents are simulated playing the aforementioned game. This will require detection software that can analyse the results of a simulation to identify (if present):

- Social synergy
- Anti-social and altruistic behaviour
- Social structures: These are persistent clusters that occur in the exchange networks formed from the interactions between agents.
- Competition between social structures
- Stable power distributions: A power hierarchy of agents that lasts for multiple rounds. This can take the form of repression, where certain agents have considerably lower amount of power compared to the others, democracy, where all agents have a similar amount of power, or dictatorship, where power is concentrated between a few agents.

This detection software can analyse many different combinations of agents to determine the particular combinations for which interesting social phenomena emerges from.

4 Designing the simulation

The simulation environment will act as a server coordinating important events occurring in the game. Agents will rely on the environment to determine when they can interact, the possible interactions that can be picked, to receive information about other agents, to improve their competency values, and to acquire more wealth through the process of mining. This approach was chosen to centralise the requests from agents to deal with the concurrent nature of the simulation and is shown in more detail in the diagram below.

Figure 4. A sequential diagram (UML) designing how agents will interact with the environment

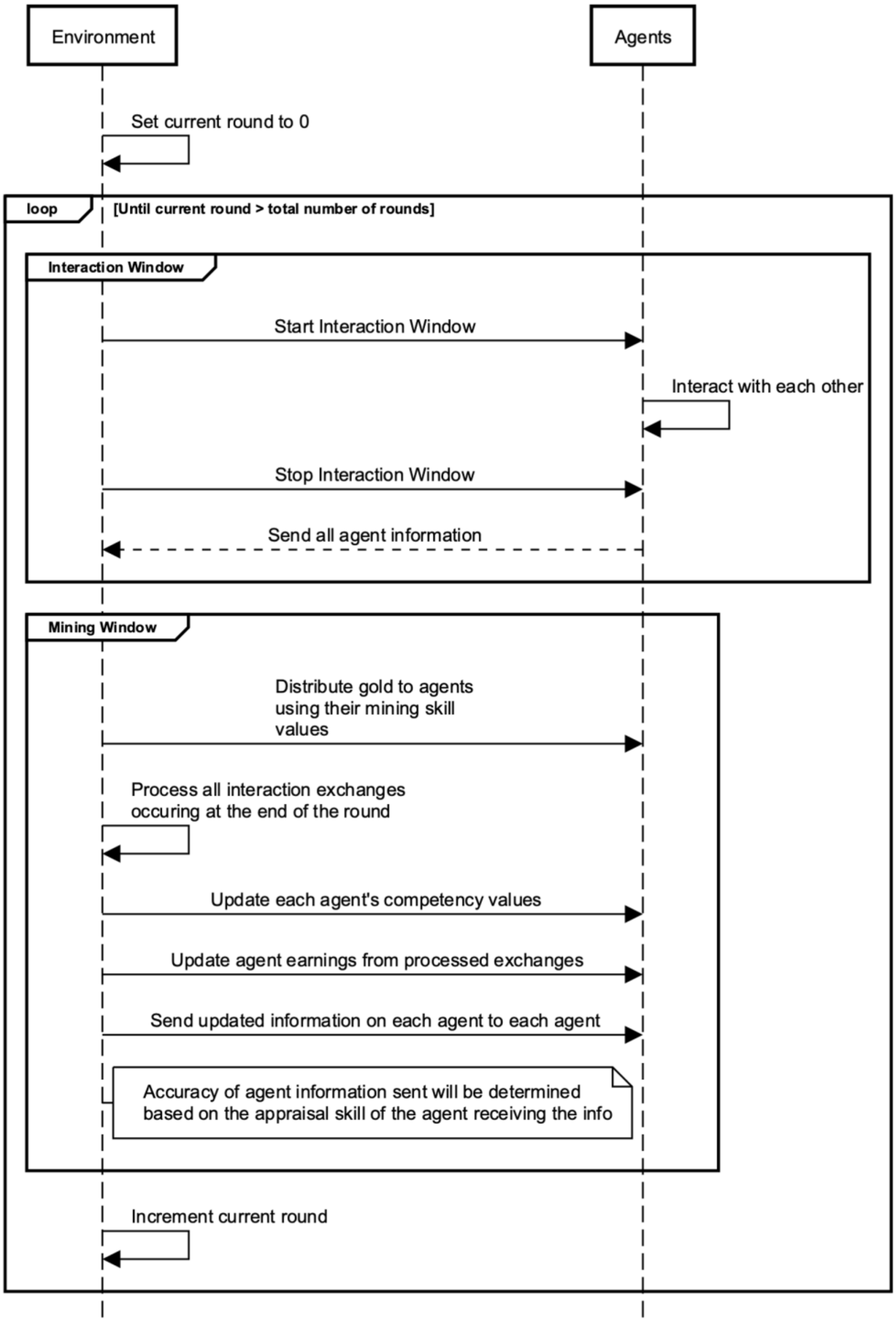


Figure 5. Sequential diagram (UML) showing agents joining the environment

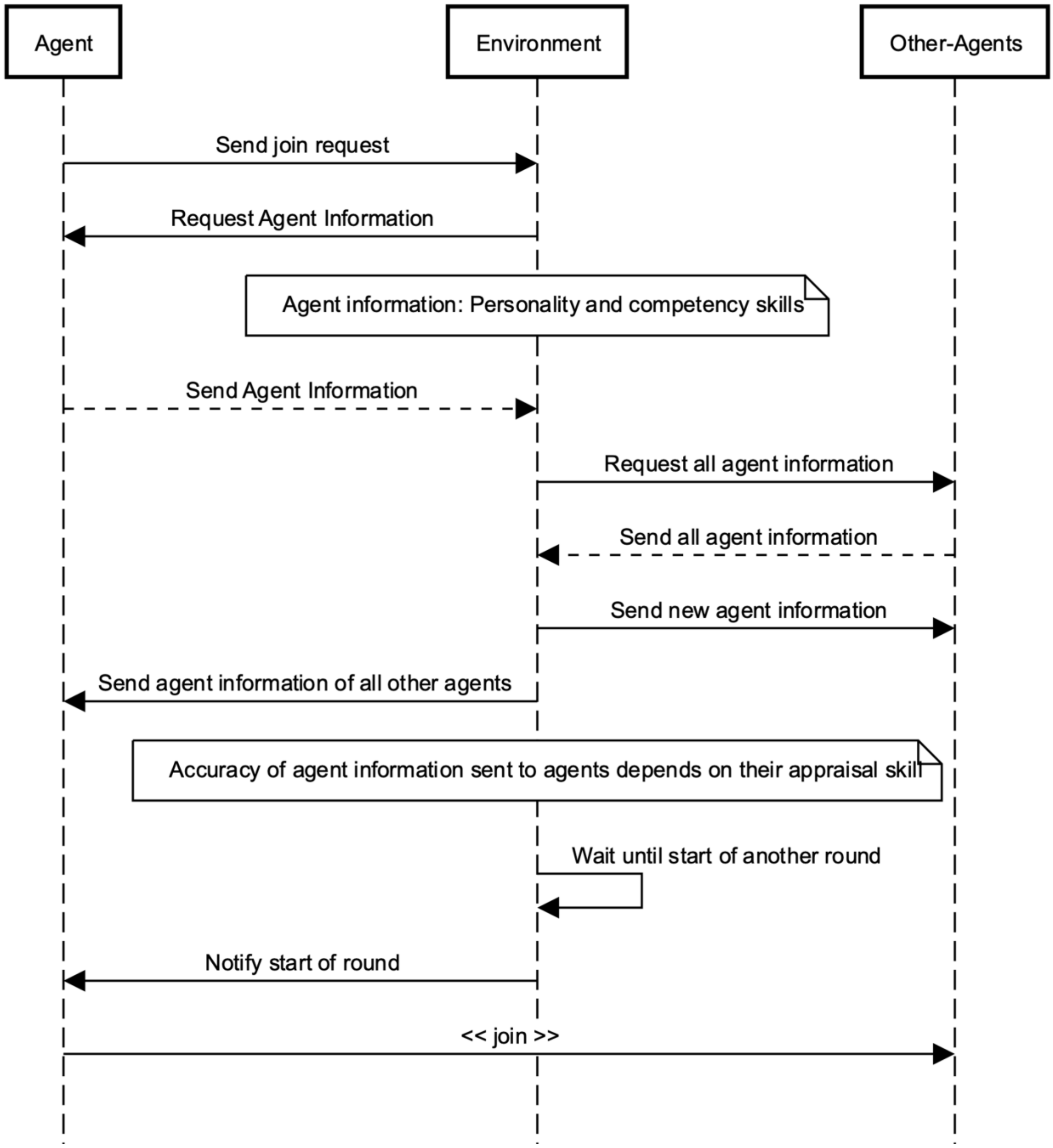
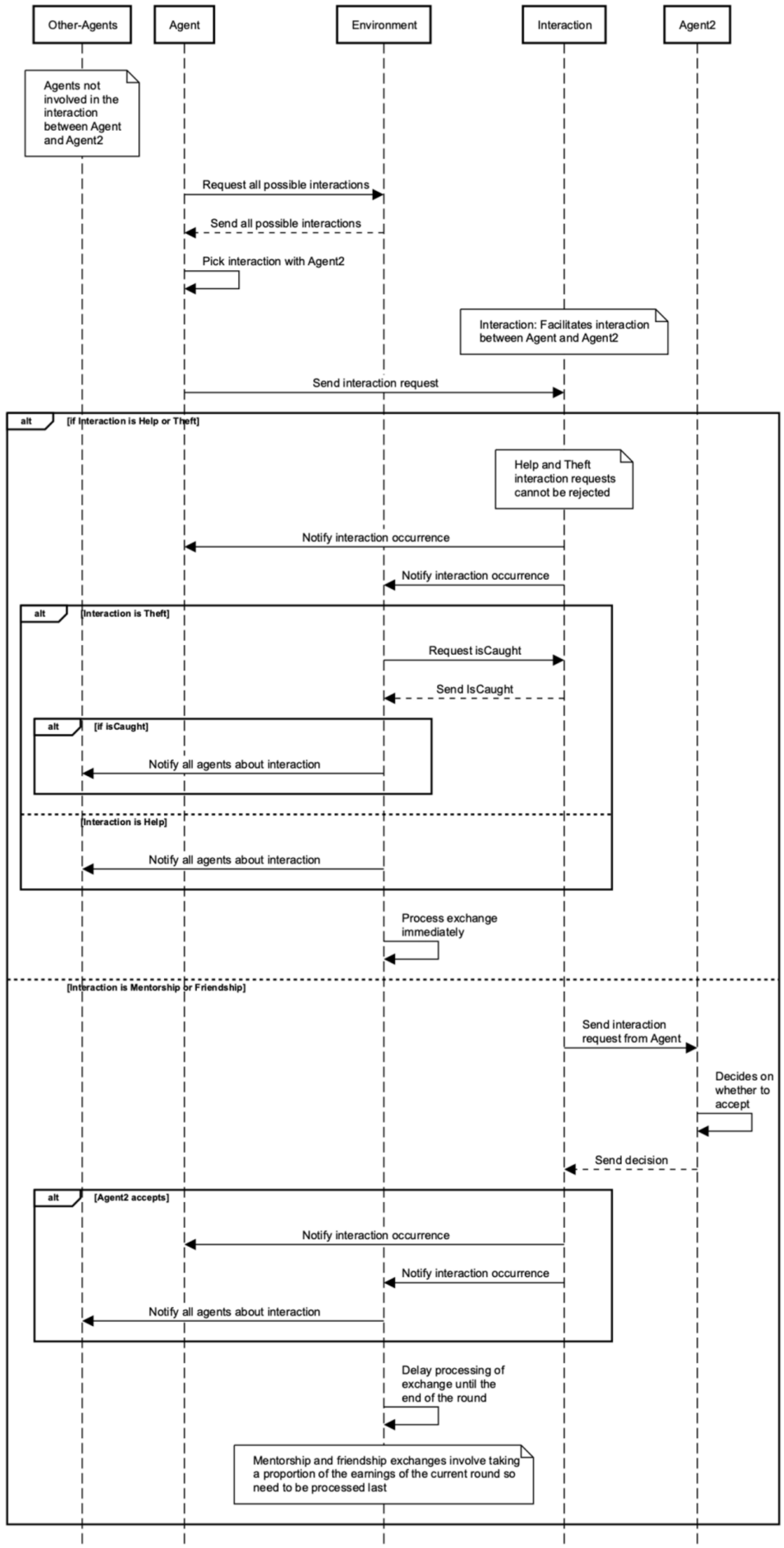


Figure 6. Sequential diagram (UML) designing the handling of interaction request

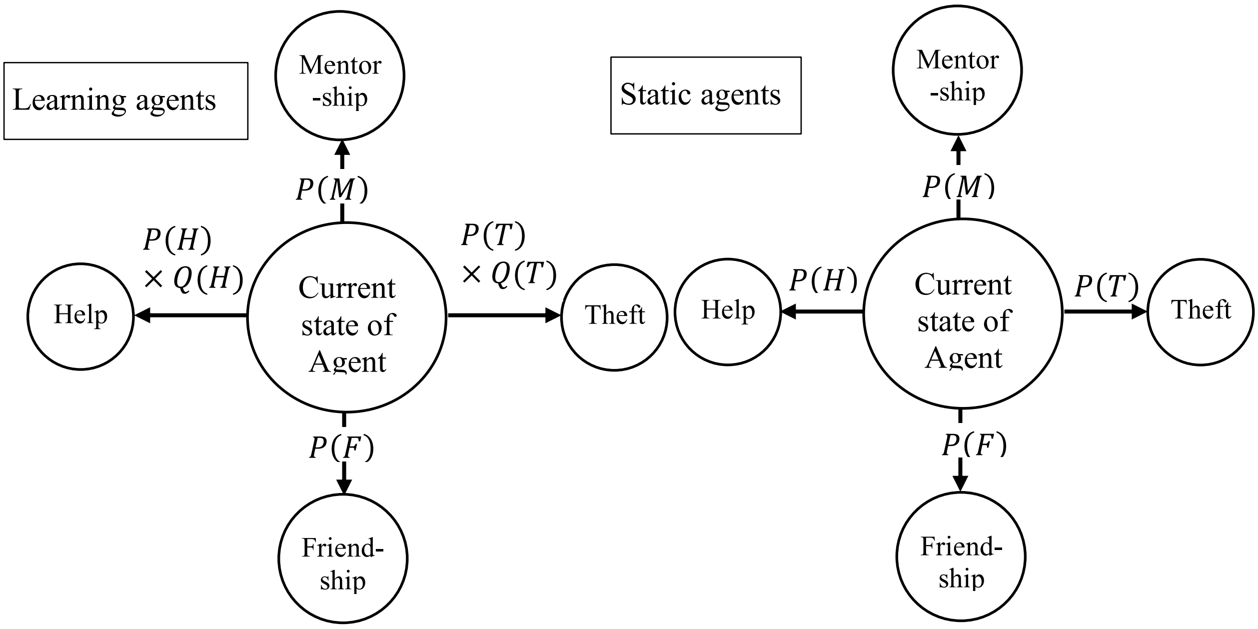


4.1 Agent Design

For this simulation, Static and Learning are the two types of agents have been designed. The assumptions made in **figure 3** will have a large influence on the behaviour seen by the agents and will be used by both types of agents to generate a probability distribution of the interaction choices. The way in which this probability distribution is used is what differentiates static and learning agents. Static agents will simply use this distribution to randomly decide what action to take. However, learning agents use Q reinforcement learning to select the most optimal choice at each point. The probability distribution will be used in the Markov decision process defined within the Q-learning mechanism to partly influence the choices made. The difference between these two approaches is illustrated below in **figure 7.**

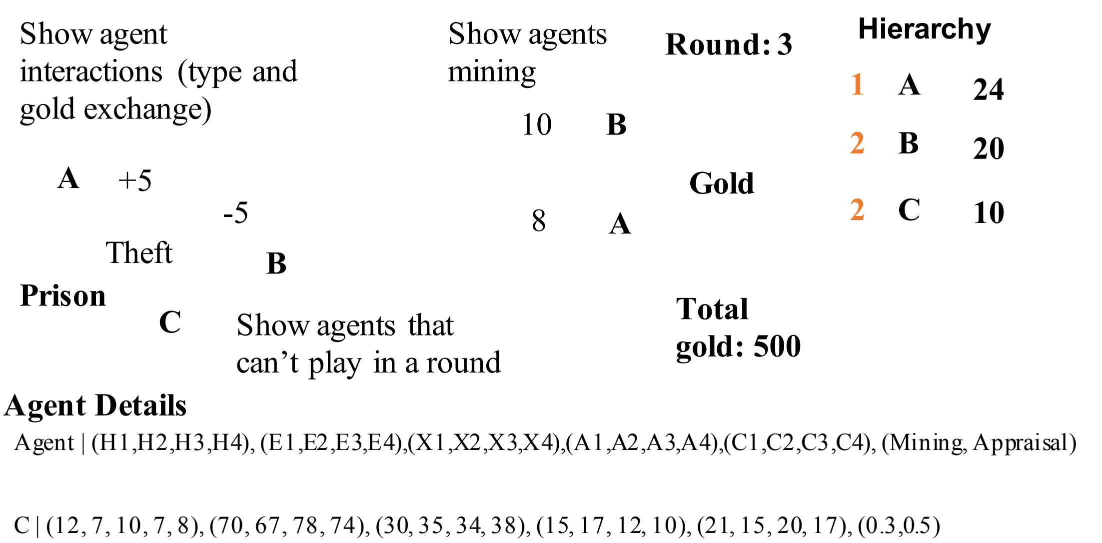
Figure 7. Decision process of agents deciding on picking one of the following interactions with another agent: Theft (T), Friendship (F), Mentorship (M) and Help (H).

$P(action) = probability\ of\ action\ occuring, Q(action) = quality\ of\ action\ (Q\ leanring)$



4.2 GUI Design

Figure 8. The preliminary design of the GUI is shown below



Since there is likely to be many interactions happening at the same time, having a visualisation of agents playing the game will be especially useful for testing and gaining a better understanding of the simulation.

4.3 Experimental Design

4.3.1 Belbin Agents

The process of experimenting to find emergent properties that can be categorised as social phenomena involves simulating with many different combinations of agents. Given that each agent has 7 continuous variables (Mining, Appraisal and 6 Hexaco personality dimensions), the total number of agent combinations will be incredibly large. Belbin team roles can be used to reduce this total. They are defined as nine clusters of human behaviour that have been researched and found to be essential in the formation of a highly successful team. Since each team role correlates to certain combinations of facets and dimensions in the Big Five Factor model (Marvin and Lindgren, 1997), they can be partially represented as agents.

Figure 9. Mappings from Belbin team roles to agent properties

Role	Description	Competence	Personality
Resource Investigator	Finds useful ideas for the team	No mappings	I, -III
Team worker	Actively encourages team synergy	No mappings	II
Co-ordinator	Team leader	**High appraisal**	IV
Plant	Creative free thinkers	No mappings	+V, -III
Monitor Evaluator	Strategic and impartial adviser	No mappings	V, -II
Shaper	Driven, focused and keeps team moving	No mappings	I, -II
Implementer	Relied upon for getting work done	No mappings	III, -V
Complete Finisher	Essential for quality control of work done	No mappings	III, -IV
Specialist	Knowledgably about key areas	No mappings	No mappings
I = Extraversion, II = Agreeableness, III = Neuroticism, IV = Openness to experience			

4.3.2 Experiment method

The purpose of the experiment is to find combinations of agents that result in social phenomena when simulated. Below shows a general outline of the steps involved in this exploratory experiment.

1. Let k be the set of agents mapped from the Belbin team roles
2. Pick N combinations of agents in set k
3. For each combination picked:
    a. Run the simulation at least 3 times
    b. Pass the results to the social phenomena detection software.
4. Create new combinations from the existing combinations where social phenomena can be detected or repeat from step 1 in the event that no social phenomena can be found
5. Repeat step 3 to find the best combination for generating social phenomena

5 IMPLEMENTATION:

Figure 10. The mathematical formula(s) corresponding to the key assumptions in **figure 3** used to create probability distribution of the different possible interaction choices.

Assumption	Formula
4	$Hexaco_P(D) = combined\ score\ percentage\ of\ all\ dimension\ \&\ facets\ in\ D$ $f(x,y) = computes\ how\ friendly\ agent\ y\ is\ to\ agent\ x$ $A = number\ of\ friend\ interactions\ between\ x\ and\ y$ $B = HexacoPersonalitySimilarity(x,y)$ $C = number\ of\ help\ interactions\ between\ x\ and\ y$ $D = number\ of\ theft\ interactions\ between\ x\ and\ y$ $E = number\ of\ caught\ thefts$ $F = number\ of\ mentorship\ interactions\ between\ x\ and\ y$ $G = total\ number\ of\ help\ interactions$ $\boldsymbol{Variable\ Importance\ in\ f(x,y)}: D > A > C > B \geq F > G \geq E$ $n = number\ of\ interaction\ rounds$ $p = number\ of\ agents$ $f = Hexaco_P(A1_x)$, $A1_x = Agreeableness: forgiveness$ $f(x,y) = 0.5 + \frac{1}{40}\left(2B + \frac{1}{n}\left(10A + 5C + 2F + \frac{G}{p}\right) - (1-0.9f)(\frac{17D}{n} + \frac{E}{p \times n})\right)$ $0 \leq f(x,y) \leq 1,\ \ f(x,y) > 0.5 \rightarrow Friends$
7	$stealing_{aversion}(x,y) = aversion\ of\ agent\ x\ to\ stealing\ from\ y\ percentage$ $help(x,y) = probability\ of\ agent\ x\ helping\ agent\ y$ $help(x,y) = \frac{wealth(x)}{2 \times wealth(y)} \times stealing_{aversion}(x,y)$
9	$theft(x,y) = probability\ of\ agent\ x\ stealing\ from\ agent\ y$ $theft(x,y) = \left(1 - stealing_{aversion}(x,y)\right) \times theft_{cost}(x) \times \left(\frac{wealth(y)}{wealth(x)}\right)$
10	$interact(x) = probability\ agent\ x\ will\ interact$ $interact(x) = \frac{1}{4}(Hexaco_p(O_x) + 3 \times Hexaco_p(X_x))$ $O_x = Openness, X_x = extraversion$
11	$Teachable(x) = \frac{1}{2}(Hexaco_p(O2, O3, H4, C) - Hexaco_p(E2))$ $O2 = inquisitiveness, O3 = creativity, H4 = modesty, E2 = anxiety$
13	$cost(x) = 1 - \frac{mentorship_{cost}}{wealth(x)}$ $0 \leq accept_{mentor}(x,y) \leq 1$ $mentor(x) = how\ desirable\ agent\ x\ is\ as\ a\ mentor$ $mentor_{want}(x,y) = \frac{1}{2}\left(mentor(y) + friend(x,y)\right)$ $accept_{mentor}(x,y)$ $= competency_{difference}(x,y) \times cost(x) \times mentor_{want}(x,y)$
14	$accept_{mentee}(x,y) = \frac{1}{3}(teachable(y) + mentor(x) + friend(x,y))$ $0 \leq accept_{mentee}(x,y) \leq 1$

See Appendix G for complete list of formulas

5.1 Static Agents

5.1.1 Characteristics

The agents will be constructed given competency and personality values. Competency is stored as decimal percentages for mining and appraisal skills. Personality is represented as a mapping from each facet (e.g. H1, H2, …,) to a percentage represented as an integer value ranging from 1 to 100.

5.1.2 Memory

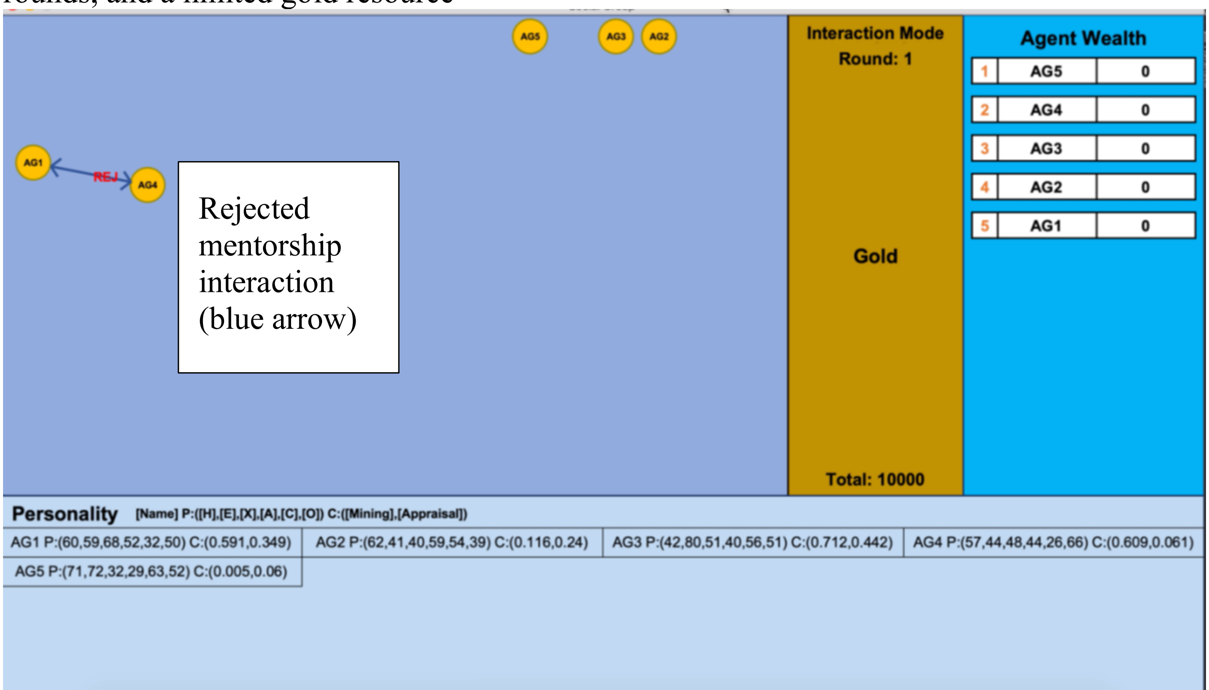
In one variable the agent stores all the interactions it has had with other agents. The agent information variable stores the following information about each of the other agents: wealth, personality, competency, total number of rounds they have both played in, and a list of all of the completed interactions observed from this agent.

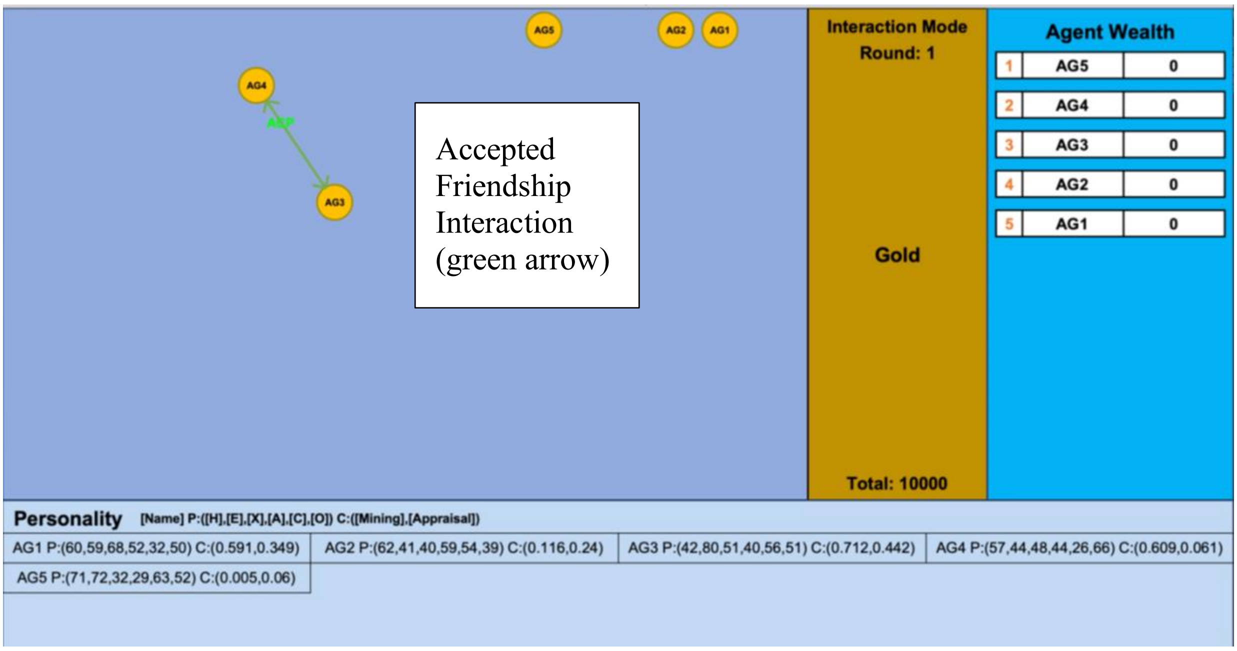
5.1.3 Behaviour

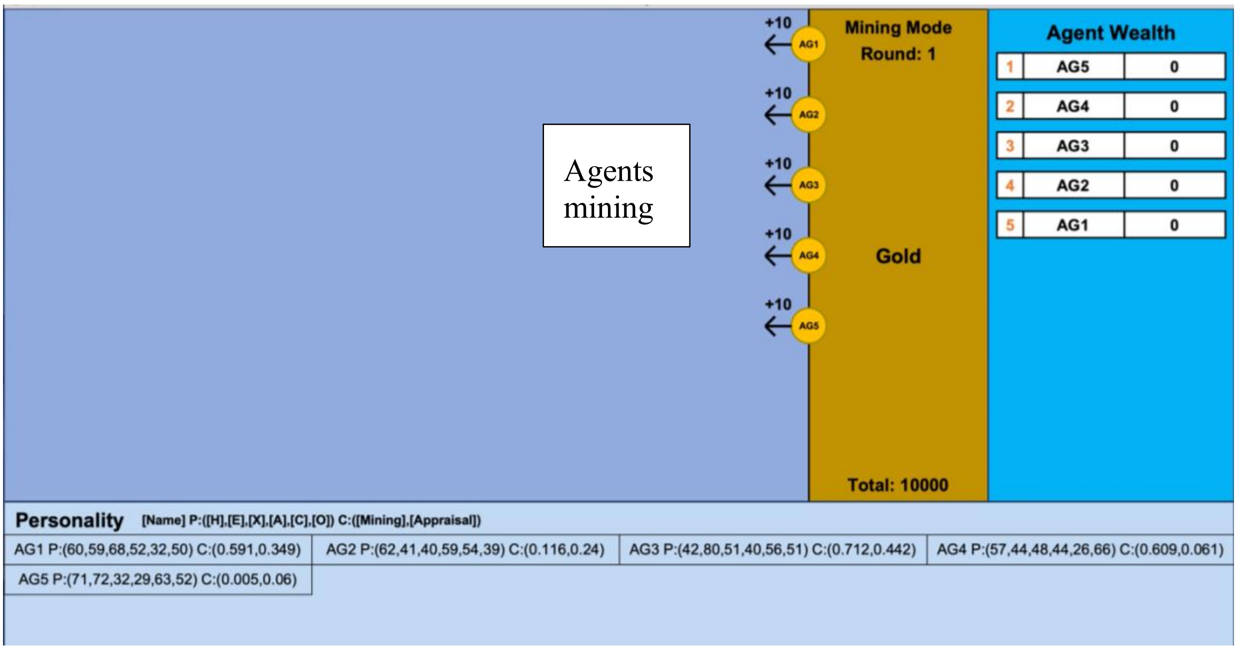
Each agent will run on its own thread separate from the main environment thread. An agent will wait for the environment to notify them of the start of the interaction window at which point, if the agent isn't in 'prison', it will decide whether it wants to interact. This decision to interact is made randomly, where the probability of interacting is determined by calling the implemented version of the interact function **in figure 10**. Upon deciding to interact, the agent will send a request to the environment to get all of the possible types of interactions with other agents as well as the corresponding probability of picking that interaction. These probabilities will be determined by the agent's request_interaction_probability(interaction) method, which are calculated using the implemented versions of the functions in **figure 10.** The static agent will use this probability distribution to randomly pick an interaction to request.

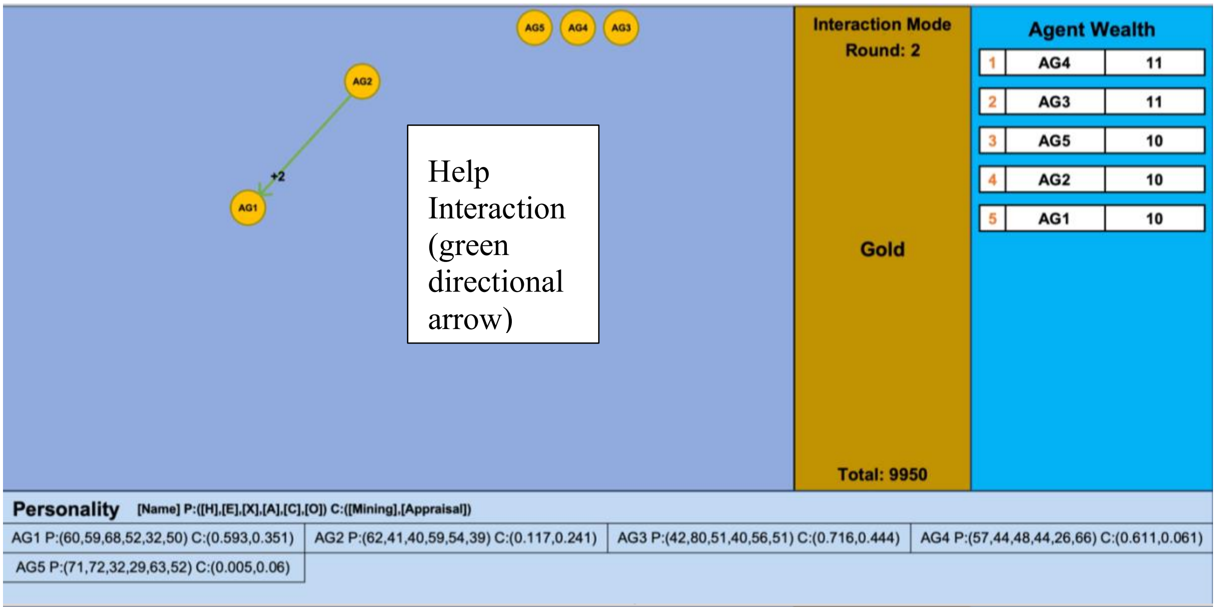
5.2 GUI

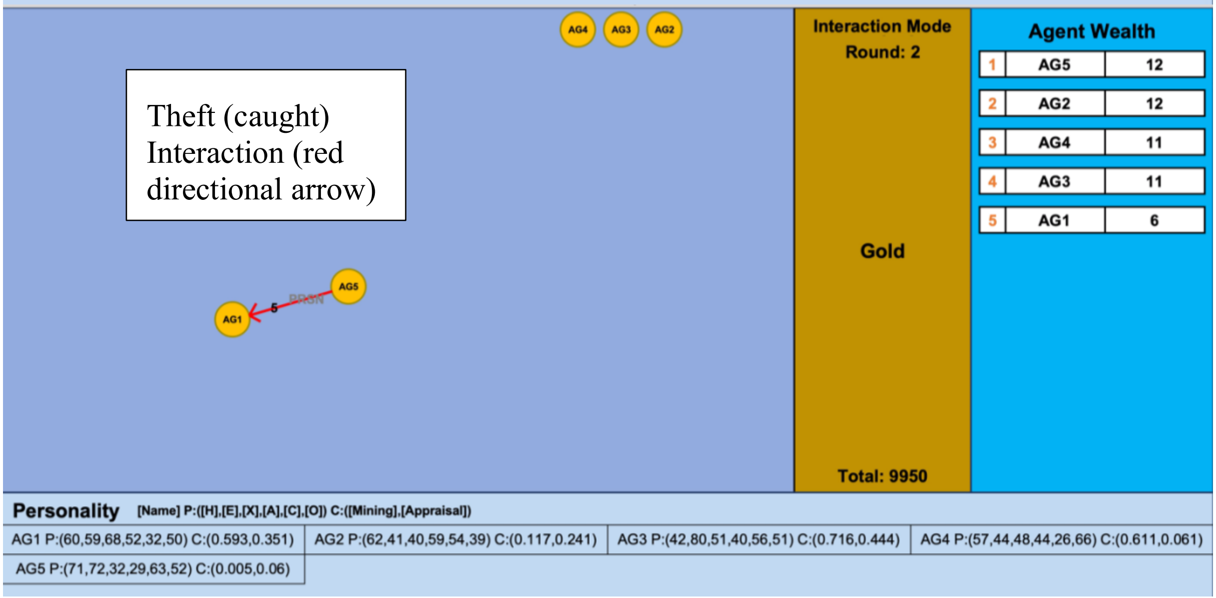
Figure 11. A set of images showing the GUI display taken from a simulation of 5 agents, 5 rounds, and a limited gold resource

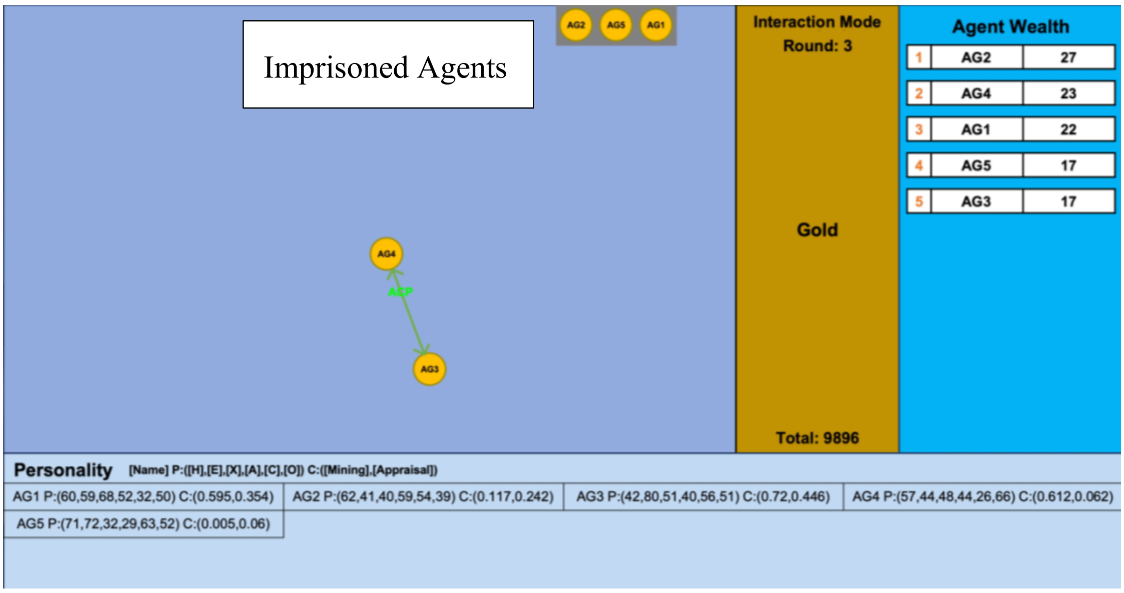



AG5
AG2
AG1
AG4
AG3
Accepted Friendship Interaction (green arrow)
Interaction Mode
Round: 1
Gold
Total: 10000
Agent Wealth
1 AG5 0
2 AG4 0
3 AG3 0
4 AG2 0
5 AG1 0
Personality [Name] P:([H],[E],[X],[A],[C],[O]) C:([Mining],[Appraisal])
AG1 P:(60,59,68,52,32,50) C:(0.591,0.349)
AG2 P:(62,41,40,59,54,39) C:(0.116,0.24)
AG3 P:(42,80,51,40,56,51) C:(0.712,0.442)
AG4 P:(57,44,48,44,26,66) C:(0.609,0.061)
AG5 P:(71,72,32,29,63,52) C:(0.005,0.06)


Agents mining
+10
AG1
+10
AG2
+10
AG3
+10
AG4
+10
AG5
Mining Mode
Round: 1
Gold
Total: 10000
Agent Wealth
1 AG5 0
2 AG4 0
3 AG3 0
4 AG2 0
5 AG1 0
Personality [Name] P:([H],[E],[X],[A],[C],[O]) C:([Mining],[Appraisal])
AG1 P:(60,59,68,52,32,50) C:(0.591,0.349)
AG2 P:(62,41,40,59,54,39) C:(0.116,0.24)
AG3 P:(42,80,51,40,56,51) C:(0.712,0.442)
AG4 P:(57,44,48,44,26,66) C:(0.609,0.061)
AG5 P:(71,72,32,29,63,52) C:(0.005,0.06)


AG5 AG4 AG3
AG2
+2
AG1
Help Interaction (green directional arrow)
Interaction Mode
Round: 2
Gold
Total: 9950
Agent Wealth
1 AG4 11
2 AG3 11
3 AG5 10
4 AG2 10
5 AG1 10
Personality [Name] P:([H],[E],[X],[A],[C],[O]) C:([Mining],[Appraisal])
AG1 P:(60,59,68,52,32,50) C:(0.593,0.351)
AG2 P:(62,41,40,59,54,39) C:(0.117,0.241)
AG3 P:(42,80,51,40,56,51) C:(0.716,0.444)
AG4 P:(57,44,48,44,26,66) C:(0.611,0.061)
AG5 P:(71,72,32,29,63,52) C:(0.005,0.06)


AG4 AG3 AG2
Theft (caught) Interaction (red directional arrow)
AG5
AG1
Interaction Mode
Round: 2
Gold
Total: 9950
Agent Wealth
1 AG5 12
2 AG2 12
3 AG4 11
4 AG3 11
5 AG1 6
Personality [Name] P:([H],[E],[X],[A],[C],[O]) C:([Mining],[Appraisal])
AG1 P:(60,59,68,52,32,50) C:(0.593,0.351)
AG2 P:(62,41,40,59,54,39) C:(0.117,0.241)
AG3 P:(42,80,51,40,56,51) C:(0.716,0.444)
AG4 P:(57,44,48,44,26,66) C:(0.611,0.061)
AG5 P:(71,72,32,29,63,52) C:(0.005,0.06)


Imprisoned Agents
AG2 AG5 AG1
AG4
AG3
Interaction Mode
Round: 3
Gold
Total: 9896
Agent Wealth
1 AG2 27
2 AG4 23
3 AG1 22
4 AG5 17
5 AG3 17
Personality [Name] P:([H],[E],[X],[A],[C],[O]) C:([Mining],[Appraisal])
AG1 P:(60,59,68,52,32,50) C:(0.595,0.354)
AG2 P:(62,41,40,59,54,39) C:(0.117,0.242)
AG3 P:(42,80,51,40,56,51) C:(0.72,0.446)
AG4 P:(57,44,48,44,26,66) C:(0.612,0.062)
AG5 P:(71,72,32,29,63,52) C:(0.005,0.06)

5.3 Testing simulation environment

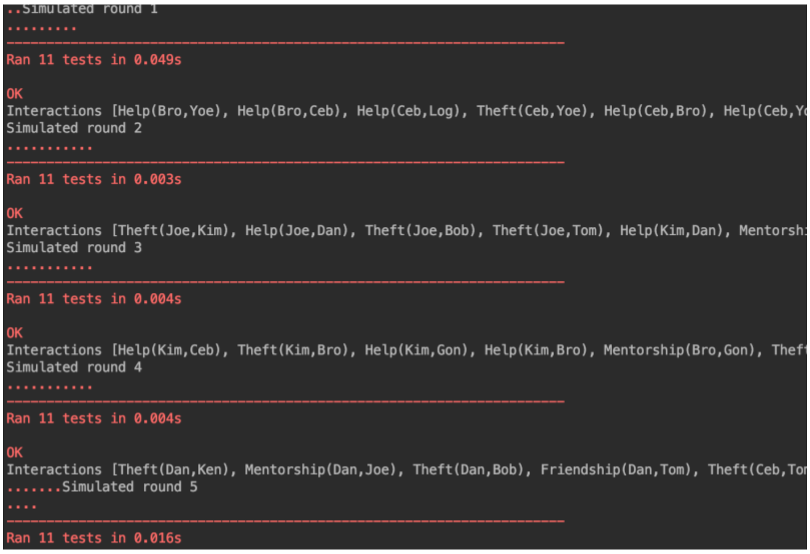
5.3.1 Test

The simulation is a dynamic system so testing has to be carried out at multiple points in the execution of the simulation. A testing strategy was defined for each rule. This testing strategy was converted to a test suite of unit tests. For each occurring interaction, each agent was checked if it has been notified. Also agents are tested to see if they have been notified on the wealth of an agent following an increase. See appendix B for detailed test strategy.

Figure 12. The method called at the end of a round to implement rule and notification testing.

```
def run_test_on_test_variables(self):
    if self.should_test:
        print("Running tests at end of round")
        # Run all tests for at the end of the interaction window
        testInteraction = TestingInteractionSuite(
            …)
        # Run all tests planned for at the end of the round
        testRound = TestingRoundSuite(
            …)
```

Figure 13. Results of running the tests



5.3.2 Assumption checking

Time was spent on checking if the assumptions on figure 3 were upheld. The purpose of this was to better understand the game by finding unexpected behaviours, which either point to an interesting property or an error. A table of expected correlations was derived from these assumptions. Some unexpected behaviours were found and identified as errors. The table in the appendix is the resulting table when the experiment was run again after fixing the errors.

5.4 Learning Mechanism

The LearningAgent is a subclass of Agent implemented to introduce 'intelligent' agents to the simulation. These agents have two modes of operation, training and working, represented by a Boolean member variable called is_training, which is passed as an argument to the constructor.

In Q-learning, the quality of an action taken to move to a state, Q(s,a), is calculated using:

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s,a,s')Q(s',a)$$

$$P(s,a,s') = probability\ of\ chooing\ action\ a\ to\ move\ from\ states\ s'to\ s$$

$$\gamma = discount\ factor$$

$$R(s,a) = reward\ of\ taking\ action\ a\ from\ state\ s$$

The purpose of the temporal difference is to calculate the quality while accounting account for changes in the environment.

$$temporal\ difference = TD(a,s) = \left(R(s,a) + \gamma \sum_{s'} P(s,a,s')Q(s',a)\right) - Q(s,a)$$

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha TD_t(a,s)$$

$$\alpha = learning\ rate\ = how\ quickly\ robot\ adapts\ to\ changes\ in\ environment$$

$$Q_t(s,a) = current\ Q\ value$$

$$Q_{t-1}(s,a) = previous\ Q\ value$$

These equations are taken from (Paul, 2020)

5.4.1 Representing the states and actions within the game

Since agents interact with other agents, the states identified for a specific agent playing the game will need to incorporate both the non-fixed states of this agent, including its competency values and the current round, and the states of the other players. This of course has the potential to be an extremely large number of states and therefore a large q matrix.

In this game, the actions a player can take are constrained to actions relating to deciding which interaction to pick. For the sake of simplicity, decisions on order of interaction will be removed from the agent's action space. The action space will only reflect, for each possible agent B, the following 9 actions an agent A can take including: accepting or rejecting B as a mentee; accepting or rejecting B as a mentor; befriending B; rejecting friendship with B; stealing from B; helping B; ignoring agent B.

The corresponding state space for agent a will therefore need to reflect:

$$a.competency, a.round,\ \ [\forall\ b \in AGENTS\ .\ (b.agent_{variable})]$$

$$AGENTS = set\ of\ all\ agents$$

$$b.agent_{variables} = (b.Hexaco(H,E,X,A,C,O), b.competency(Mining, Appraisal))$$

Instead of storing each agent variable as a separate dimensions to the array, they have been represented as two dimensions to save memory space. Due to this change, the quality of an action with another agent X will have to be evaluated for each agent variable value of X. One of the problems of using this shortened q matrix is the extra processing required both in the learning and choosing stages to set or access the quality of a particular action.

Extract: Shows how q-matrixes are stored in LearningAgent.py

```
# Q matrix:[competency state: 10][round state: 10][other-agent variables:
8][other-agent variable level: 5][interaction choice: 9]
# q_array_sizes = [10, 10, 8, 5, 9]
self.Q = np.array(np.zeros(q_array_sizes))
```

5.4.2 Reward Function

$$\boldsymbol{R}(\boldsymbol{a},\boldsymbol{s},\boldsymbol{i}) = \frac{gold_{earnt}(a,i)}{gold_{earnt}(a,s)} + \sum_{s \le round \le s+effect} discount^{round-start} * \frac{gold_{earnt}(a,round)}{total_{earnt}(round)}$$

$s = round\ when\ interaction\ i\ happened$

$effect = number\ of\ rounds\ for\ which\ interaction\ should\ still\ be\ rewarded$

$0 < discount < 1, \qquad a = agent, \qquad i = interaction$

$gold_{earnt}(agent, interaction) = gold\ earned\ by\ agent\ in\ interaction$

$gold_{earnt}(agent, round) = gold\ earned\ by\ agent\ in\ round$

$total_{earnt}(round) = total\ earnt\ by\ all\ agents\ in\ round$

During training, when an agent carries out an action by requesting, accepting or rejecting an interaction with another agent, the action and the probability of it being taken are stored as pending lessons. The quality of the action will be evaluated and updated for each agent variable of the other agent after a pre-determined amount of rounds (effect) take place. The purpose of counting other round earnings in the reward function is to consider the long term consequences.

5.4.3 Choosing Actions

When the Learning Agent is in working mode, i.e. is_training variable is false, its behaviour when requesting, accepting or rejecting an interaction changes from being the same as static agents to using the evaluated q-matrix to determine the optimal action.

Quality value of an interaction with agent B for agent A:

$$Q(A,i) = \sum_{(avar,value) \in B.agent_{variables}} QM[A.competency][A.round][avar][value][i]$$

$B.agent_{variables} = [\ (Extraversion, 10\%), \dots, (Mining, 50\%)$

$QM = q\ matrix$

To determine which interaction to request, learning agents will find the highest quality interaction for each agent it’s playing against, using the above formula, and will use the quality value of each of these interactions as weights to make a random choice – if the average quality is below a certain threshold, no interaction will be requested. The purpose of incorporating randomness to this decision is to emulate the irrational and curious nature of people. Determining whether to accept an interaction is much simpler as it just uses the quality values of accepting or rejecting an interaction to decide.

5.5 Training

The first step of training the Learning Agents involved generating agents that represent the Belbin roles. For each Belbin role, three levels (low, medium and high) were defined for the mining competency and two levels (low, high) were defined for the appraisal competency, the only exception being the coordinator role where a single level was used for appraisal. This generated 43 different training agents; for example, CO*L*H, CO*M*H, CO*H*H is generated for just the coordinator role.

5.5.1 Training Environment:

The games were played with an unlimited amount of gold in the environment, 40 rounds, a starting mining amount of 10 and with 12 players each round. These settings were chosen to maximise the number of agents that can be trained and the number of rounds for each game while minimising the time taken for each simulation to run since many simulations will be required for training.

5.5.2 Training parameters:

- Discount (reward function) of 0.85
- Gamma (evaluating q) of 0.9
- Alpha (evaluating q) of 0.9
- Effect (reward function) of 5 rounds

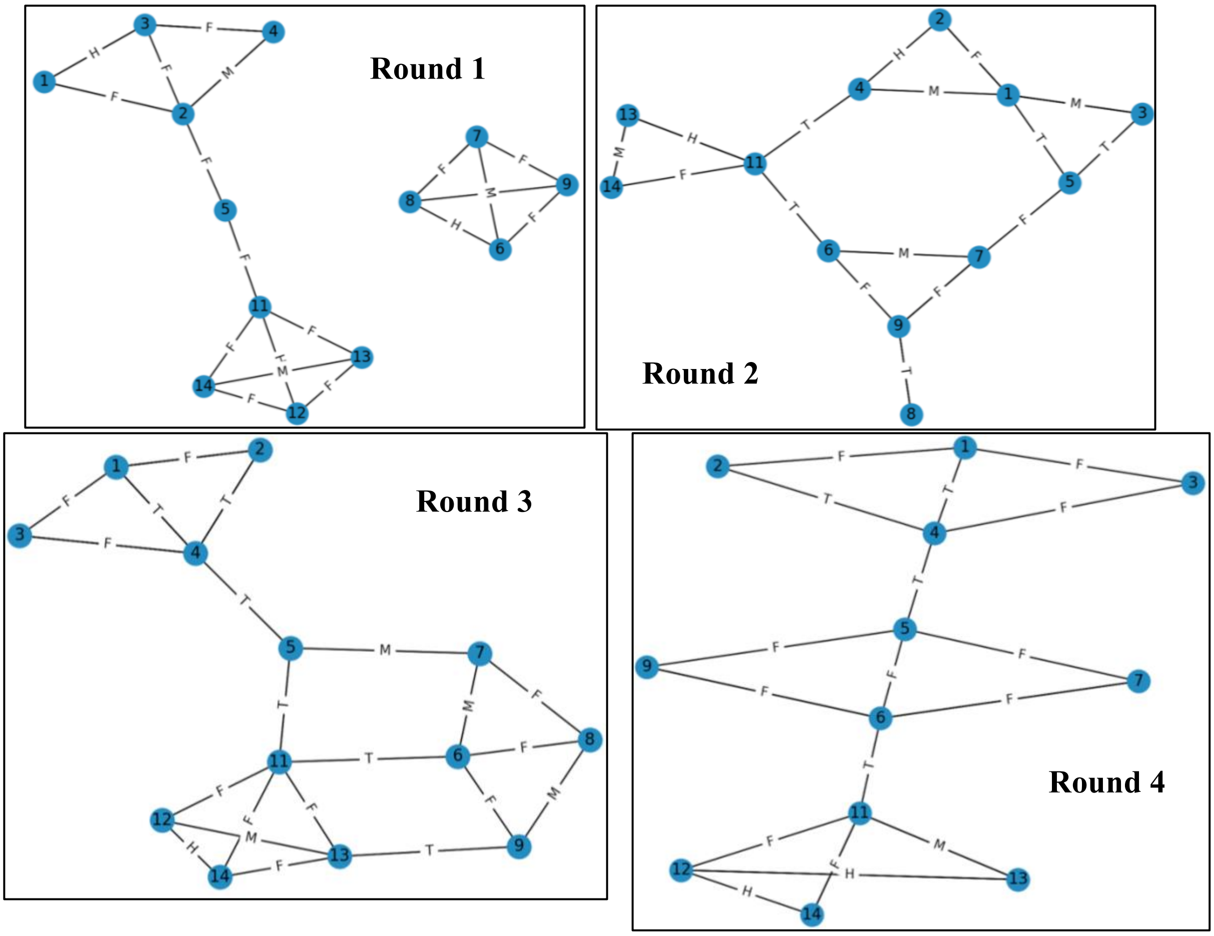
In each game 6 LearningAgents and 6 static agents were picked out of the 43 agents for training. Training was set up to ensure that a random set of the least picked LearningAgents and static agents was picked each time to maximise the variation of agents trained against. Since learning requires a substantial amount of processing for each agent, a limit of 6 agents was chosen to reduce the simulation time. A total of 3179 games were simulated for training allowing for each agent to be trained for at least 75 games.

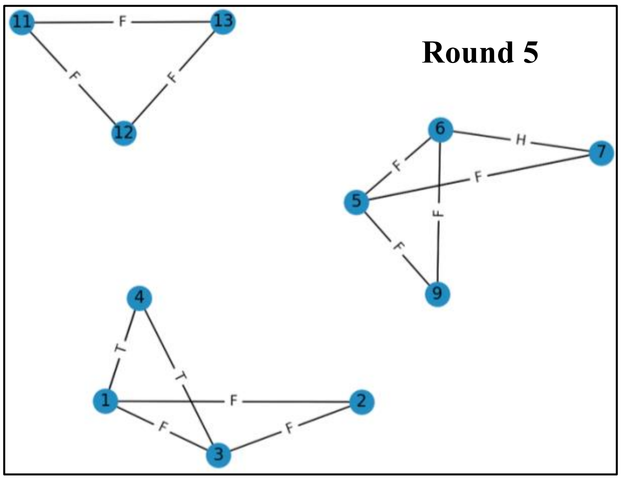
5.6 Finding social phenomena

5.6.1 Finding Social Structures

To find social structures, an exchange network was created for each round of interactions. An exchange network is a graphical representation showing all of the exchanges of gold between each agent for each interaction. The detection software was fed a map of each round to a list of interactions, and it started by converting each list of interactions to a networkx (python module) graph, representing the exchange network for that round. Using the asynchronous label propagation algorithm implemented in the networkx module, communities of agents, which are groupings of densely connecting nodes in the exchange network, were identified for each round of interactions. The communities of consecutive interaction rounds were compared and only the pairs of communities with a similarity percentage over 80% were kept. Consecutive chains of these pairs of communities were identified to find persistent social structures occurring in the game and only the chains that lasted for more than 3 rounds in a row were returned by the find_social_structures method.

Figure 14. The exchange networks and corresponding social structure identified by the detection software when tested on a simulation of agents playing for 5 rounds





Identified Social Groups:

1. [9, 6, 7, 8] from 1 to 3
2. [2, 4, 3, 1] from 2 to 5
3. [11, 14, 12, 5, 13] from 3 to 5

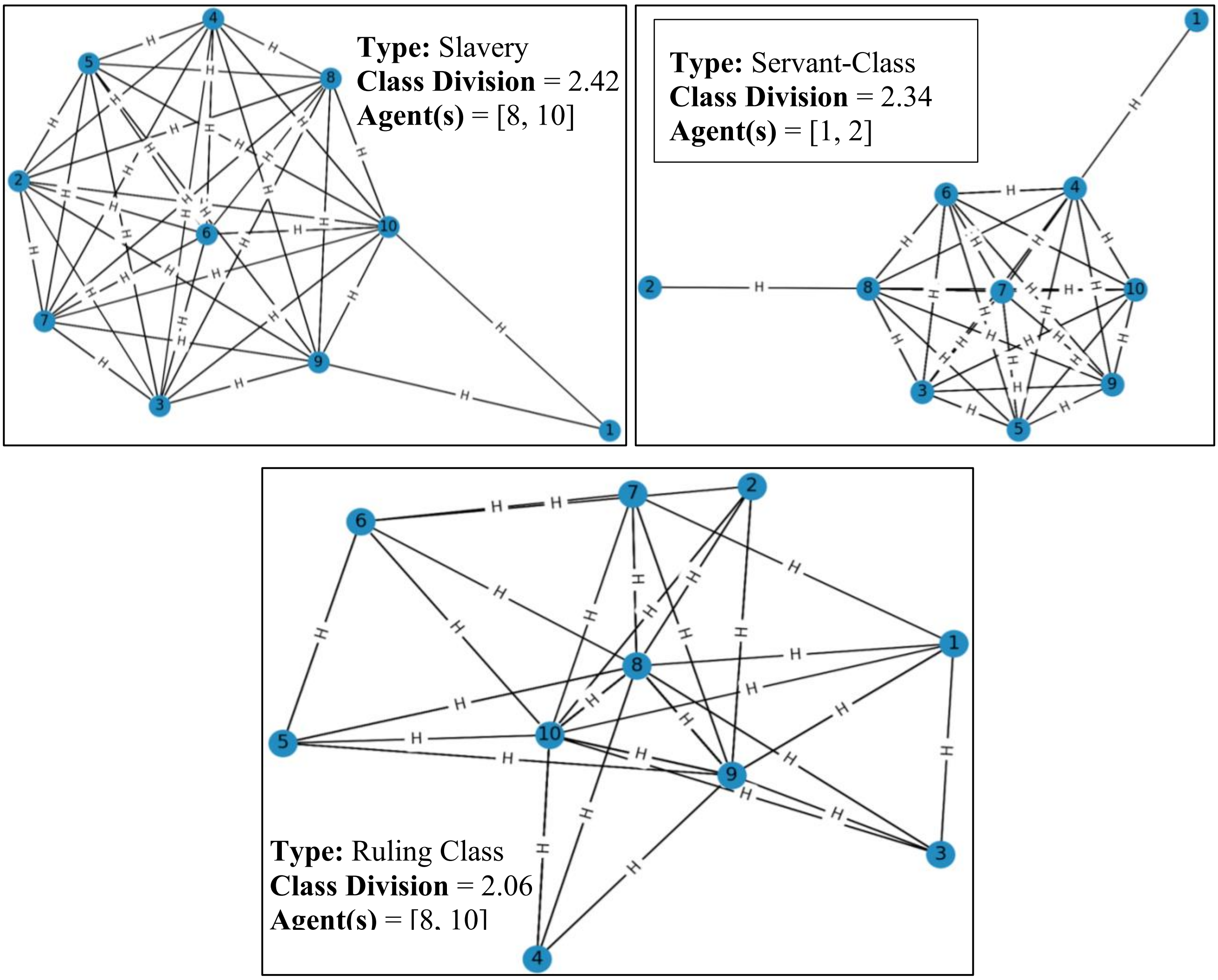
5.6.2 Finding Power Distributions

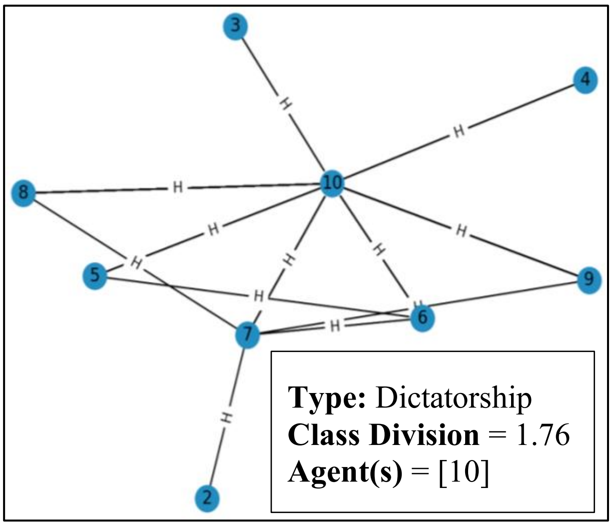
The research done by Cook and Emerson 1978 showed that power in some exchange networks is linked to the point centrality of the network. In the implementation, the power_distribution method of the detection software will return the power of each agent when given a set of interaction and a map of agents to earnings. To do this it constructs an exchange network from the interactions and uses the closeness centrality algorithm (networkx) to find the point centrality of each agent. The power of an agent is calculated as the average of its point centrality and its earnings.

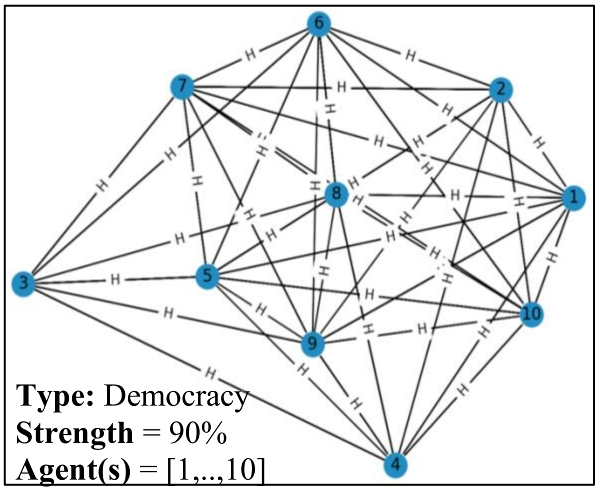
Type of distribution

Democracy: Identified when the standard deviation divided by the mean of the distribution is less than 0.2.	
A class division, implying that the distribution of data is split into multiple groupings, can be identified by calculating the distance from the mean divided by the standard deviation of the distribution for each agent's power in the distribution and finding the difference between average positive value and the average negative one. If this difference is greater than a threshold (1.5 was chosen) then a class division has been identified. A ruling class is identified when there are fewer values with positive distances (from the mean) than negative ones, and a servant class is identified when there are fewer negative ones.	
Ruling Class: All of the agents with a power value greater than x, where x is the largest power value subtracted by 20% of the standard deviation.	Servant-Class: All of the agents with a power value less than x, where x is the smallest power value added by 20% of the standard deviation.
Dictatorship: A special ruling class consisting of just one agent	Slavery: A special servant class consisting of just one agent

Figure 15. The exchange networks and identified power distribution type information generated by test suite







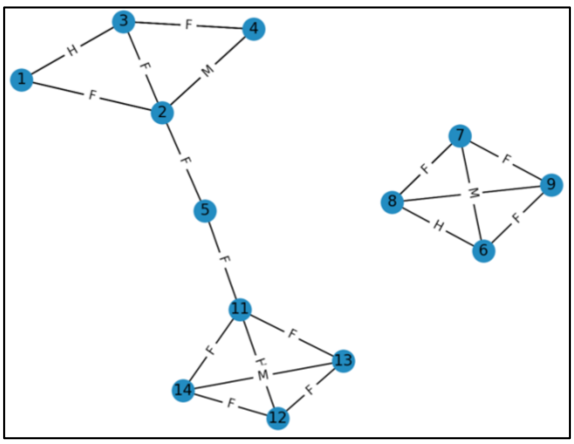
The Power-Stability class within the detection software can analyse the power hierarchy stability across multiple rounds, when supplied with the agent earnings and interactions for each round. This class uses the Power-Metrics class, to calculate the power distribution for each round of interactions and to identify their type and strength. This data will be used to find the most stable power hierarchies, or in other words, the hierarchies that persist across multiple rounds. For each hierarchy found, it will return the stability percentage, which is proportion of rounds it has been maintained, average strength, and a list of all of the agents involved. A similar method has been used to determine the wealth stability.

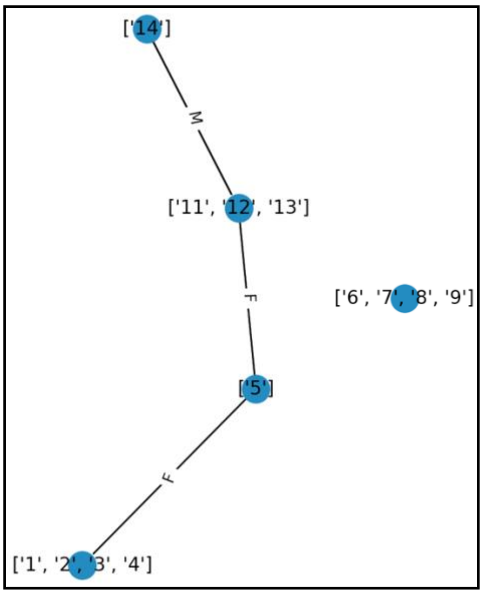
The Social-Metrics class inside the detection software extracts interaction and agent earning information from the simulation to return analysis data in the form of a Social-Metrics object

Figure 16. The Schema of the Social-Metrics object

Data	Description
Anti-social level	Percentage of earnings from theft interactions
Cooperation level	Percentage of earnings from help interactions
Friendliness	Percentage of earnings from friend interactions
Productivity	Percentage of earning from mentorship interactions
Social synergy	Total amount of additional gold introduced to the environment

Figure 17. Exchange network with social structures as nodes





5.6.3 Competition between Social Groups

The detection software will also analyse competition amongst social groups by calculating the social metrics of interacting social groups. This is accomplished by treating these groups as a single node in an exchange network (see figure 17); any interaction occurring from an agent in one social group to an agent in another social group is treated as an interaction between those two social groups.

5.6.4 Experiment

Upon completion of a simulation, the detection software will return a Social Analysis Result data object to encapsulate all of the social phenomena detected. The schema of this object can be found in the appendix.

In the implemented version of the experiment detailed in page X, an experimental run consists of the following steps:

1. Input a set of combinations of Belbin Learning Agents
2. For each combination of agents
    - Run a simulation for this combination three times, generating 3 different Social Analysis Result data objects
    - Merging these data objects into one by taking the average and similarity across the three simulation runs for each relevant data point
    - Storing each merged data point on Excel readable CSV files
3. For each agent contained involved, find the averages for each relevant data point and store on Excel readable CSV files.

The social structures identified across the 3 simulation runs for each combination will be assigned a consistency score, which is the percentage of social structures identified in all of these runs. The main experiment involves trying many different experimental runs, and for each one, looking in the spreadsheet to determine if any social phenomena has been found.

5.6.5 Testing Detection Software

As referenced earlier, a testing suit was created in TestingSocialAnalysis.py to thoroughly test all classes associated with the detection software. The first 10 unit tests were created to test social structure detection, power distribution categorisation, power and wealth hierarchy stability, social metrics calculations and social group competition analysis. The last 4 are integration tests that fully test the processes defined for an experimental run by running multiple simulations for various, predefined, combinations and checking to see if the Social Analysis Result data objects returned have been correctly merged and saved in the appropriate excel files.

Figure 18. The results of running the Testing Detection Software testing suit

```
Launching unittests with arguments python -m unittest TestingSocialAnalysis.TestingDetectionSoftware

Ran 14 tests in 0.006s

OK
```

6 Chapter 6: Results

6.1 Experimental Run #0

Initially, a set of 32 different combinations of 12 agents, each assigned a single Belbin role, a mining competency level and an appraisal skill level (e.g. CO*MH or ME*LL) was generated for the experiment. With this experiment, there are many different possible combinations of 12 agents so the specific combinations chosen are of little importance in the initial search provided that they meet the conditions: all combinations are unique and a variety of different competency values included for a variety of different Belbin roles.

Figure 19. Frequency in the initial set of 32 combinations	
Belbin Roles	'RI': 48, 'CO': 48, 'CF': 48, 'TW': 48, 'P': 48, 'I': 48, 'ME': 48, 'S': 48
Mining	'L': 128, 'M': 128, 'H': 128
Appraisal	'L': 192, 'H': 192

On completion of this experimental run, none of the Learning Agents showed any power hierarchy stability, they all had a power stability percentage of 0%, and not a single social structure could be identified from any of them.

Figure 20. The 12 agents with the highest average social synergy		
Agent	Social-Synergy Mean	Social-Synergy Similarity
I*HL	248	82.6
S*MH	246	83.1
P*HH	244	76.7
RI*HH	242	79.9
I*MH	242	82.8
TW*LL	241	77.4
CF*ML	236	69.8
CF*HH	234	82.5
S*HL	234	82.6
TW*LH	232	75.5
TW*ML	232	84.8
CO*MH	231	72.9

6.2 Experimental Run #1

The next set of combinations used were constructed from these high synergy agents to trigger social phenomena. A set of 10 combinations was chosen in such a way to ensure that each combination is unique and that each agent, in the entirety of the set, is included 10 times.

Figure 21. The combinations with stable power hierarchies									
	Overall		Democracy		Servant Class		Slavery		None
Combination	SP	SIM	SP	S	SP	S	SP	S	SP
CF*HL CF*HL I*HL I*HL I*HL ME*MH ME*MH P*HH P*HH TW*HL TW*HL TW*HL	3.41	57	0.85	27	0.85	54	1.7	66	0
CF*HH CF*HH CF*HL CF*HL I*HL I*HL ME*MH ME*MH P*HH P*HH TW*HL TW*HL	3.41	13	0	0	0	0	0	0	3.41
CF*HH CO*MH CO*MH I*HH I*HH RI*HL RI*HL S*LH S*LH S*MH TW*ML TW*ML	0.85	0	0	0	0.85	51	0	0	0
SP = Stability (%), SIM = Similarity of SP across 3 runs (%), S = Strength (%)									

Only two combinations out of the possible 10 showed some stability, albeit extremely minor, in the power hierarchies. However, this was an improvement to the initial combination runs where no stability was identified. Additionally, this set of combinations led to the formation of social structures, as seen in figure 23. The ones that formed have a consistency percentage of 0% meaning that they only emerged in one of the 3 simulation runs.

Figure 22. All of the agents found in identified social structures				
Agent	Total Groups	Social-Synergy Mean	Wealth-Stability Mean (%)	Power Stability Mean (%)
I*HL	5	669	49.1	1.71
TW*HL	5	667	48.9	1.71
CF*HL	4	559	52.1	1.37
P*HH	4	559	51.1	1.37
ME*MH	4	519	52.9	1.37
I*HH	0	467	48.9	0.171
CF*HH	2	462	50.7	0.769

As seen in figure 22, the total group number correlates with social synergy and power stability.

Figure 23. The combinations where social structures were found		
Combination	Consistency-Score	Social Structures
CF*HL CF*HL I*HL I*HL I*HL ME*MH ME*MH P*HH P*HH TW*HL TW*HL TW*HL	0	[[CF*HL I*HL ME*MH P*HH TW*HL]:**9**]
CF*HH CF*HH CF*HL CF*HL I*HL I*HL ME*MH ME*MH P*HH P*HH TW*HL TW*HL	0	[[CF*HH I*HL ME*MH P*HH TW*HL]:**3**]

The social group of CF*HL, I*HL, ME*MH, P*HH, and TW*HL was the only social structure that emerged from the set of 10 combinations of high synergy agents. The numbers 9 and 3 indicate the number of rounds (out of 120) for which this social structure has been identified.

Figure 24.				
Combination	Anti-Social	Productivity	Cooperation	Friendliness
CF*HH CF*HH CF*HL CF*HL I*HL I*HL ME*MH ME*MH P*HH P*HH TW*HL TW*HL	61.8%	11.3%	16.2%	10.7%

The interactions between the social groups identified for the first combination were highly anti-social (figure 24), which is normally indicative of competition amongst the social structures. However, due to the fact that the average anti-social percentage for each combination in this run is 57.8% with a standard deviation of 3.2%, this level of anti-social behaviour doesn't indicate an elevated level of social structure competition.

6.3 Experimental Run #2

Given that there was some success in finding social phenomena from high synergy agents, the next set of combinations was generated purely from the highest synergy agents identified from experimenting with the previous set of combinations; the fact that these agents also were the ones with the highest average power stability and greatest presence in social group further incentivised this approach.

As expected, more social phenomena emerged from these combinations. **Figure 25** shows that a greater amount of power hierarchy stability can be identified from more combinations in this run compared to the previous ones. Furthermore, like with the previous runs, there is a greater inclination towards repression and democratic power distributions. Moreover, more consistent social structures have emerged from these combinations compared to the previous runs, as seen on **figure 26.**

Figure 25. The combinations with stable power hierarchies

	Overall		Stability percentage (%)			
Combination	SP	SIM	DM	DC	SC	SL
CF*HL CF*HL I*HH I*HH I*HL I*HL ME*MH ME*MH P*HH P*HH TW*HL TW*HL	0.85	0	0	0	0	0.85
CF*HH CF*HH CF*HH CF*HL CF*HL CF*HL I*HL I*HL I*HL TW*HL TW*HL TW*HL	9.4	37	5.1	0.85	0	3.4
I*HH I*HH I*HH I*HH ME*MH ME*MH ME*MH ME*MH P*HH P*HH P*HH P*HH	36	93	36	0	0	0
CF*HH CF*HH CF*HL I*HH I*HL I*HL ME*MH P*HH TW*HL TW*HL	1.7	13	0	0.85	0	0.85

CF*HH CF*HH CF*HL CF*HL I*HL I*HL ME*MH ME*MH P*HH P*HH TW*HL TW*HL	1.7	0	0	0	0.85	0.85
CF*HH CF*HL CF*HL I*HH I*HL ME*MH ME*MH P*HH P*HH TW*HL	0.85	0	0	0.85	0	0
DM = Democracy, DC = Dictatorship, SC = Servant-Class, SL = Slavery, SP = Stability Percentage (%), SIM = similarity of SP across 3 simulation runs of that combination						

Figure 26. The combinations where social structures were found

Combination	Consistency-Score	Consistent Social-Structures	All Social Structures
CF*HH CF*HH CF*HH CF*HL CF*HL CF*HL I*HL I*HL I*HL TW*HL TW*HL TW*HL	33%	[CF*HH CF*HL I*HL TW*HL]	[[CF*HH CF*HL I*HL TW*HL]:15]
I*HH I*HH I*HH I*HH ME*MH ME*MH ME*MH ME*MH P*HH P*HH P*HH P*HH	0	[]	[[I*HH ME*MH P*HH]:10]
CF*HH CF*HH CF*HL I*HH I*HH I*HL I*HL ME*MH P*HH TW*HL TW*HL	0	[]	[[CF*HH CF*HL I*HH I*HL ME*MH P*HH TW*HL]:4]

Similar to the previous run, the interactions between these social structures do not indicate an elevated level of competition amongst social structures

7 Findings

The goal of this project was to investigate whether social phenomena could be identified by applying rational exchange theory to agents interacting in a social environment. The results show that simple social phenomena has emerged from the simulations.

In terms of social structure formation, the results have shown that when multiples of the higher synergy agents are grouped exclusively together, social structures will more consistently emerge from the simulation. The social structures that were formed could only be identified when there were multiples of agents for each Belbin representation; when the constituent agents for these structures were placed, without repetition of Belbin representation, in plenty of other groups with a myriad of different agents, no social structures emerged. Additionally, there weren't any emergent social structures consisting solely of agents with the same information. This suggests that while these agents are more likely to form social structures in the presence of similar agents, they prefer to interact with agents more different than themselves. While this behaviour is certainly an example of organisational social phenomena, out of the many combinations simulated, not a single one could be found with social structures that clearly exhibited collective action behaviour. As mentioned earlier, the social metrics identified in figure 24 is close to the average metrics identified from each combination. With more complex social structures, this distribution is likely to deviate from this average; for example, with insular social structures, there will be a smaller percentage of interactions between other social structures, with highly competitive social structures, there will be a greater percentage of earnings from anti-social interactions between other groups, and with highly sociable structures, there will be a lower percentage of anti-social earnings.

In regards to power hierarchy formation, the results indicate that stable power hierarchies can form between these agents in the simulation. Additionally, the most stable and consistent hierarchies tended to emerge from combinations of agents, where social structure formation could also be identified. Unlike with social structure phenomena, this phenomena can more consistently be identified when simulated multiple times. Although stable power hierarchies have been shown to emerge in some simulations, the simulations, where they do form, are highly unstable for at least 64% of the entire run. Finally, the results show that these stable power hierarchies tend to take the shape of more recognisable power distributions such as repression and democracy and so, even considering the instability of the simulations, the fact that these types of power distributions can even be observed and a link has been established between the power stability of hierarchies and the formation of social structures provides evidence of complex social phenomena occurring from simulating the social exchange theory.

8 Critical Evaluation

While these results do indicate that the goal of finding social phenomena from simulating the exchange theory has been achieved, there are some limitations to this project that affects the validity of these results.

One of the limitations of this research lies in the limited amount of training afforded to the ‘intelligent’ agents. Although the training of these agents always presented a challenge due to the large number of available states and actions, this difficulty has been exacerbated by the approach to train all 43 agents derived from the Belbin roles. Having said this, this approach is not without merits since it allows for more unique combinations of agents to be explored resulting in a more expansive search for social phenomena. Furthermore, having more agents with a variety of different competencies and personalities compete in this simulation will be more reflective of society and this is extremely desirable for simulating the social exchange theory. In this case, there’s a trade-off between choosing to train specialist agents more effectively allowing for more ‘intelligence’ and so better emulations of people and choosing a generalist approach, which better emulates society by incorporating more diversity. However it can be said that choosing to train 43 agents is ambitious even considering this generalist approach therefore it would have been more fitting to have used fewer variations of competency when creating the Belbin agents.

	Anti-socialness		Productivity		Cooperation		Friendliness	
Agent	Mean	Similarity	Mean	Similarity	Mean	Similarity	Mean	Similarity
Learning	57.1	93.5	15.7	82.4	17.4	86.3	9.81	83.5
Static	49.8	93.3	12.7	82.6	17.5	90.9	20.0	88.8

Another limitation of this research lies in the lack of balance between the actions available in the game. The agents playing this game can steal 30% of what the victim earned in the previous round, and can continuously steal from other agents in the same round. To balance this action, agents are imprisoned and so must skip as many rounds as they have been caught stealing, and stealing from other agents will reduce the level of “friendship” between the two agents, which will have long term effects on the number of interactions they have. The fact that earnings from theft interactions alone account for the majority of the interaction earnings of the agents highlights that the previous balancing rules have failed. This lack of balance has been exploited by the Learning-Agents leading to fewer interactions (especially friendship) between them due to the resulting enmity from pursuing theft interactions. In future works, this lack of balance can be fixed by restricting the number of possible theft interactions in a given round or by altering the function calculating the probability of agents stealing in the mathematical model.

Although this unintentionally elevated level of anti-socialness potentially reduces the number of possible interactions, besides theft, this doesn’t invalidate the simulation of the social exchange theory since the increase is indiscriminate. However, this restriction potentially does affect the quality or the complexity of the social phenomena that can be observed, which may be a reason as to why no collective action behaviour has been found.

9 Project Planning

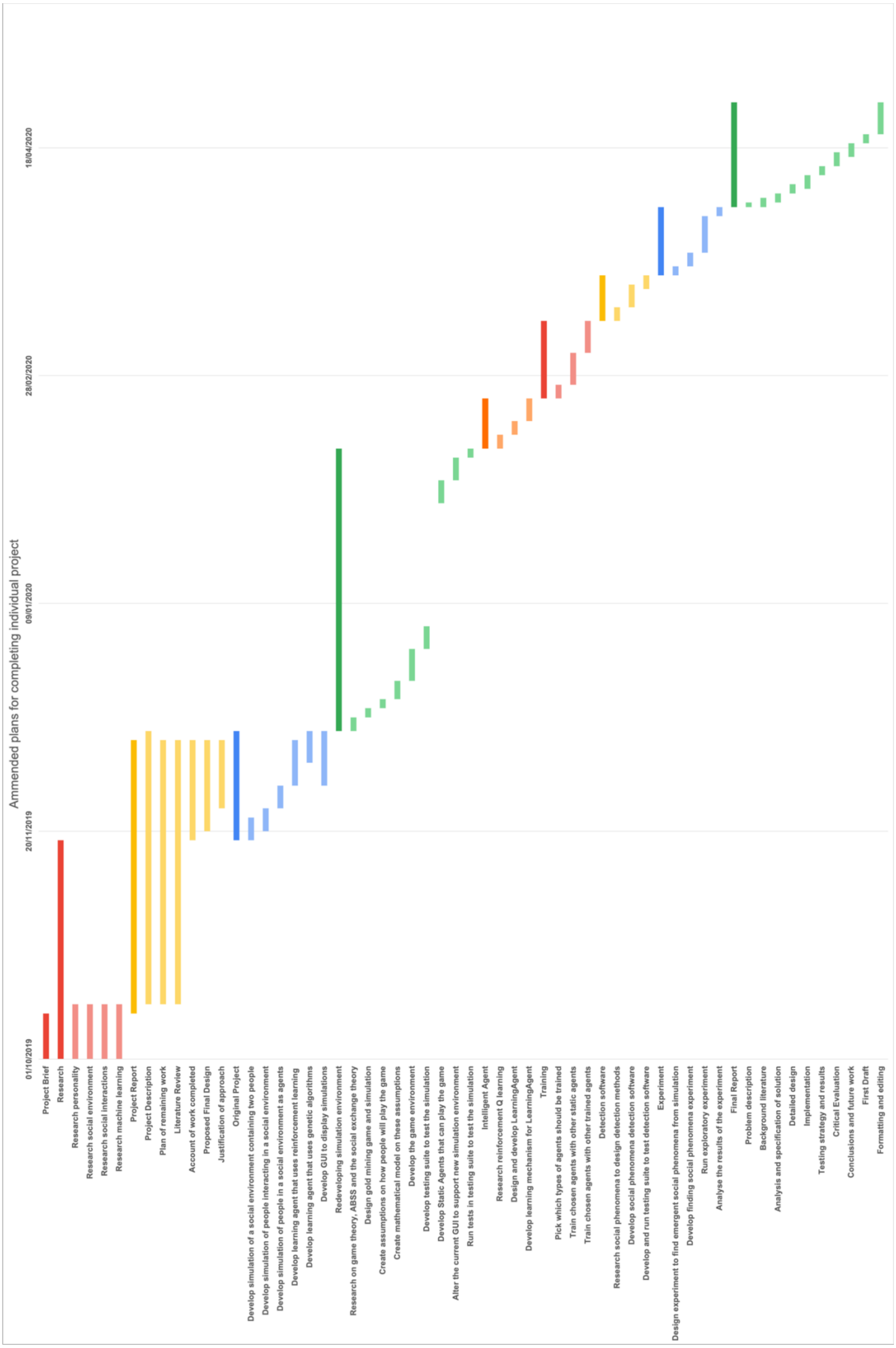
This project has evolved from the one proposed in the brief and expanded upon in the project report. Both projects involve using agent-based social simulation to simulate people interacting in a social environment however the purpose for doing so is what differentiates the two. The original project focuses on the machine learning aspect developing agents that can compete for status in a generalised social environment, whereas this project is an investigation into the resultant social phenomena of agents competing for gold in a game type environment. Once this project had evolved to the point where the plans made in the progress report were unusable, a new Gant chart **(**seen on next pages**)** was created to plan the new direction.

The direction of the project was changed because of the difficulties faced in trying to develop a generalised social environment based on the Whitworth and Sylla, 2012 model; the developed generalised model became too abstract and convoluted for any meaningful results to be found, shifting my focus onto specialised environments based on game theory instead. In the process of reading around the subject of social interaction theory, the rational choice and social exchange theory stood out as especially relevant ideas, and so these were also incorporated into the new project.

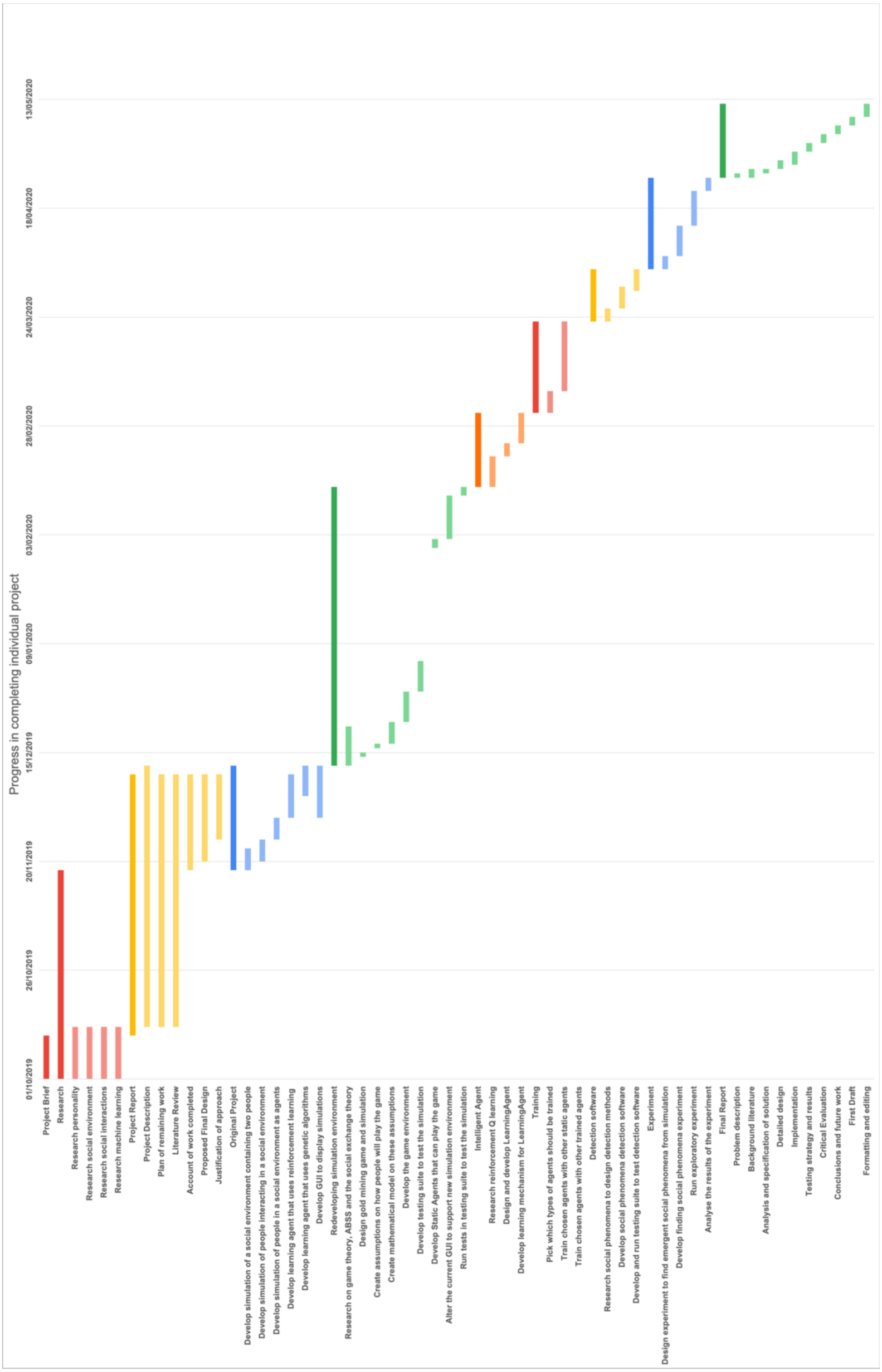
9.1 Effect of COVID-19

Due to the COVID pandemic, the lab computer facilities could not be accessed for 3 weeks so the final part of this project had to be completed using a computer with much less RAM and processing speed than was required. The time taken to run the simulations of 12 agents and 40 rounds with this computer was around 20 seconds, and this time would increase (polynomial) as the number of agents increased. To ensure the feasibility of training, simulation settings of 12 agents and 40 round, which are both less than what was originally planned, were chosen. Additionally, training the 43 agents with just static agents took much longer than anticipated, and so there wasn't any time left for another round of training with trained agents. Having had better computer facilities, more time could be allotted to training the agents resulting in better learning mechanisms and so better agents used.

1.1 Planned Schedule Gantt Chart



1.2 Actual Progress Gantt Chart



10 Conclusion

In conclusion, this project has demonstrated how social phenomena can emerge amongst agents interacting in a simulated environment through simple application of the social exchange theory and given some assumptions on the behaviour of agents. Specifically, the results show the emergence of simple social structures and stable power distributions, such as repression and democracy. Furthermore, a link between these two social phenomena has been identified, since in most of the cases, where one has been found, the other has also been found. The success of this project has been impacted by the limited amount of training of the Learning-Agents and the unbalanced rules of the game. Having said this, although the extent of learning done by the agents has been reduced, they still managed to learn strategies to, at the very least, exploit the unbalanced theft rules, indicative of a partially functional learning mechanism, which ultimately is the threshold of achievement required for this solution.

This research was done with another goal of finding insights on the how the exchange theory can support or counter its critiques. Since this is an idealised representation of the social exchange theory in process based on many different assumptions on how personality and competence influences interactions of real people, all of the emergent properties found exist only within this idealised social environment, even if they are analogous to properties of the real one. Although simple organisational behaviour and power stability has emerged from this idealised social environment, macrophenomena, where groups of agents have collective objectives that they choose over their own individual interests, has not emerged. This observation highlights the difficulties of achieving collective action only through application of the social exchange theory without the notion of selective incentives to reward such behaviour.

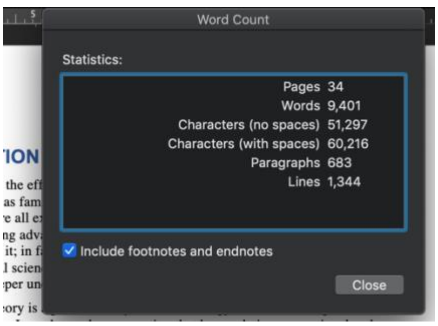
10.1 Future works:

The next step for this project is to investigate how complex social macrophenomena can occur in this simulation. It does certainly appear that explicitly defined selective incentives are required to allow for the emergence of such social phenomena; in the context of this game, this would be rewards for players who form specific interaction networks i.e. agents who form a network of mentorship interactions, or in other words a school structure, can be rewarded with compounded competency increases for mentees and extra earnings (not given by mentee) supplied to the mentors.

The following are other potential avenues for exploration using this simulation environment

- Study the effect of adding and removing agents on the stable power hierarchies and social structures formed.
- Study the effect of setting a limit to the amount of minable gold in the environment
- Adding more interactions i.e. more anti-social and pro-social interactions besides Theft and Help, respectively
- Adding more competencies besides appraisal and mining; for example, teaching and thief competencies for mentorship and theft interactions respectively.

10.2 Word Count



References

1. Anderson, C., John, O., Keltner, D. and Kring, A., 2001. Who attains social status? Effects of personality and physical attractiveness in social groups. Journal of Personality and Social Psychology, 81(1), pp.116-132.
2. Cook, K., Emerson, R., Gillmore, M. and Yamagishi, T., 1983. The Distribution of Power in Exchange Networks: Theory and Experimental Results. American Journal of Sociology, 89(2), pp.275-305.
3. Doran, J. and Gilbert, N., 1994. Simulating Societies Using Distributed AI. pp.1-18.
4. Elstad, E., Christophersen, K. and Turmo, A., 2011. Social exchange theory as an explanation of organizational citizenship behaviour among teachers. International Journal of Leadership in Education, 14(4), pp.405-421.
5. Grossi, D., Dignum, F., Dastani, M. and Royakkers, L., 2005. Foundations of organizational structures in multiagent systems. Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems - AAMAS '05,.
6. Latané, B., Nowak, A. and Liu, J., 1994. Measuring emergent social phenomena: Dynamism, polarization, and clustering as order parameters of social systems. Behavioral Science, 39(1), pp.1-24.
7. Li, T., Qiu, Y., Yue, P. and Zhong, G., 2007. Exploiting Model of Personality and Emotion of Learning Companion Agent. 2007 IEEE/ACS International Conference on Computer Systems and Applications,.
8. Liu, Z., 2008. A personality model of virtual characters. 2008 7th World Congress on Intelligent Control and Automation,.
9. Marvin, R. and Lindgren, B., 1997. R. Meredith Belbin'S Team Roles Viewed From The Perspective Of The Big 5. Oslo: University of Oslo.
10. McCrae, R. and John, O., 1992. An Introduction to the Five-Factor Model and Its Applications. Journal of Personality, 60(2), pp.175-215.
11. Paul, S., 2020. An Introduction To Q-Learning: Reinforcement Learning. [online] FloydHub Blog. Available at: <https://blog.floydhub.com/an-introduction-to-q-learning-reinforcement-learning/> [Accessed 7 May 2020].
12. Roberts, B., Kuncel, N., Shiner, R., Caspi, A. and Goldberg, L., 2007. The Power of Personality: The Comparative Validity of Personality Traits, Socioeconomic Status, and Cognitive Ability for Predicting Important Life Outcomes. Perspectives on Psychological Science, 2(4), pp.313-345.
13. Scott, J., 2020. Rational choice theory. In: UNDERSTANDING CONTEMPORARY SOCIETY: THEORIES OF THE PRESENT. London: SAGE Publications, pp.126 - 138.
14. Whitworth, B. and Sylla, C., 2012. A social environmental model of socio-technical performance. International Journal of Networking and Virtual Organisations, 11(1), p.1.
15. Zafirovski, M., 2005. Social Exchange Theory under Scrutiny: A Positive Critique of its Economic-Behaviorist Formulations. Electronic Journal of Sociology,.
16. Karimi, S. and Kangavari, M. (2012). A Computational Model of Personality. *Procedia - Social and Behavioral Sciences*, 32, pp.184-196.

17. Snellman, J., Barrio, R. and Kasaki, K., 2019. Social structure formation in a network of agents playing a hybrid of ultimatum and dictator games.
18. Basu, Sumit, Choudhury, Tanzeem, Clarkson, Brian and Pentland, Alex. (2001). Learning Human Interactions with the Influence Model.
19. Hechter, M. and Kanazawa, S. (1997). Sociological Rational Choice Theory. *Annual Review of Sociology*, 23(1), pp.191-214.
20. Cropanzano, R. and Mitchell, M. (2005). Social Exchange Theory: An Interdisciplinary Review. *Journal of Management*, 31(6), pp.874-900.
21. Selfhout, M., Burk, W., Branje, S., Denissen, J., van Aken, M. and Meeus, W. (2010). Emerging Late Adolescent Friendship Networks and Big Five Personality Traits: A Social Network Approach. *Journal of Personality*, 78(2), pp.509-538.
22. DeYoung, C., Hirsh, J., Shane, M., Papademetris, X., Rajeevan, N. and Gray, J. (2010). Testing Predictions From Personality Neuroscience. *Psychological Science*, 21(6), pp.820-828.
23. Burke, E., Kendall, G. and Sastry, K. (2016). *Search methodologies*. Springer-Verlag New York, pp.97-125.
24. Sutton, R. and Barto, A. (1998). *Reinforcement learning*. Cambridge, MA: MIT Press.
25. Li, Xiaochen and Mao, Wenji and Zeng, Daniel Dajun and Wang, Fei-Yue. (2008). Agent-Based Social Simulation and Modelling in Social Computing. Lecture Notes in Computer Science. 5075.

Appendices

Appendix. A Rules of the game

1. A single round consists of an interaction window, where agents are able to interact with each other and a mining window where agents are given resources
2. Agents are assigned resources in accordance with their mining skill: higher mining skill, means more resources assigned
3. Each agent can choose to interact with another agent during the interaction window
4. Let t = time limit for each interaction window
5. An agent is limited to $\sqrt{total\ number\ of\ possible\ interactions\ for\ agent}$ interactions each interaction round
6. Interactions can take the form of friendship, mentorship, help and theft
7. Friendship
    - Condition: No condition to friendship
    - Can be requested by any agent. Requires acceptance
    - Roles: Agent can only take one role in this interaction as a Friend so between two agents there can only exist a single friendship interaction
    - At the end of the mining window of the current round, each friend will be awarded with 10% of all of the earnings of the friend in the current round
8. Mentorship
    - Roles: Agent can take the role of a Mentor or a Mentee in this interaction
    - Condition: competency(Mentor) > competency(Mentee)
    - Can be requested by both an aspiring mentor or mentee. Requires acceptance
    - At the end of the mining window of the current round:
        - Mentor will gain 25% of all of the earnings of the friend in the current round
        - Mentee will increase competency skill by 15% of the smallest skill difference percentage from the mentor's competency
9. Help
    - Roles: Agent can take the role of a helper or beneficiary
    - Condition: wealth(helper) > help.cost
    - Only a helper can request it. It doesn't require acceptance
    - Cost to help will be 5% of the money earnt by the beneficiary in the last round or at least 25% of the minimum mining amount
10. Theft
    - Roles: An agent can either be a victim or a thief in these interactions
    - If caught agent cannot interact and will not be able to mine in the next round
    - Only a thief can request it. It doesn't require acceptance
    - Thief will steal 30% of the money earnt by victim in the last round
    - Note: Agents who are caught steal from another agent won't be taken out of the game until the end of the round. However for each time they are caught stealing, they will be imprisoned for an additional round.
11. In the interaction rounds, there can only be a single interaction, where each agent takes up different roles, for each type between two different agents i.e. If Theft(A,B) and Theft(B, A) exists then no other theft interaction can take place between A and B.
12. At the end of each round, mining skills increase in proportion to conscientiousness of agent. At the end of each round, appraisal skills increase in proportion to extraversion of agents

Appendix. B Testing strategy for Game Rules

Rules	**Strategy**
Total number of interactions less than limit for each agent	Check at end of interaction window Inputs: agents, confirmed-interactions
For any x, y, $\lvert Friendship(x,y)\rvert + \lvert Friendship(y,x)\rvert \leq 1$ and $x \neq y$	Check at end of interaction window Inputs: interactions
For any x, y, $\lvert Mentorship(x,y)\rvert \leq 1$ and $\lvert Mentorship(y,x)\rvert \leq 1$	Check at end of interaction window Inputs: interactions
For any x, y, $\lvert Theft(x,y)\rvert \leq 1$ and $\lvert Theft(y,x)\rvert \leq 1$ and $x \neq y$	Check at end of interaction window Inputs: interactions
For any x, y, $\lvert Help(x,y)\rvert \leq 1$ and $\lvert Help(y,x)\rvert \leq 1$ and $x \neq y$	Check at end of interaction window Inputs: interactions
Before any Mentorship interaction $M(a,b)$: $a.Competency < b.Competency$ and $a \neq b$	Check before mentorship interaction is accepted or when it is requested Input: Mentorship interaction
Each friend is awarded an extra 10% of friend earnings	Check at end of round Input: Friendship interactions, Earnings before, Earnings after
Mentee will increase competency skill by 15% of the smallest skill difference percentage from the mentor's competency	Check at end of interaction window Input: Skill differences of agents, Competency before, Competency after
Before any help interaction Help(a, b): $wealth(a) > Help.cost$	Check before help interaction is accepted or when it is requested Input: Help interaction
Check Theft and Help interaction exchanges are processed correctly	Check at the end of interaction round Input: Last round earnings, earnings before, earnings after, theft and help interactions
For any Help interaction (H): $H.RequestingAgent = H.ProactiveAgent$ For any Theft interaction (T): T.$RequestingAgent = T.ProactiveAgent$	Check at the end of interaction round Input: Interactions
Mining and appraisal skills increase at the end of each round	Check at the end of each round Input: Competency before, Competency after

Appendix. C Assumption checking

		Spearman		Pearson		
Independent Variable	Dependent Variable	rs	p	r	p	Expected?
Conscientiousness	Total Competency Increase	0.333	0.35	0.4638	0.1769	YSI
Conscientiousness	Mentorship\P	0.255	0.48	0.304	0.3932	YSI
Conscientiousness	Total Gold Earned	0.30	0.404	0.2708	0.4492	YSI
Mining	Total Gold Earned	1	0	0.9952	0	Y
Mining	Mentorship\P	0.741	0.022	0.7488	0.0202	Y
Mining	Mentorship\R	-0.861	0.003	-0.8978	0.001	Y
Conscientiousness	Mentorship\R	0.506	0.14	0.5847	0.0759	Y
Extraversion	Total Competency Increase	1	0	0.9988	0	Y
Extraversion	Friendship	0.985	0	0.9788	0	Y
Extraversion	Mentorship\P	0.936	0.0001	0.9288	0.0001	Y
Extraversion	Mentorship\R	0.858	0.002	0.8487	0.0019	Y
Appraisal	Mentorship\P	0.516	0.15	0.6186	0.0758	Y
Appraisal	Mentorship\R	-0.726	0.027	-0.755	0.0187	Y
Appraisal	Total Gold Earned	0.867	0.0025	0.8906	0.0013	Y
Emotionality	Total Competency Increase	-0.964	0	-0.9604	0	Y
Emotionality	Mentorship\R	-0.658	0.038	-0.6909	0.0269	Y
Honesty-Humility	Theft\P	-0.871	0.001	-0.8535	0.0017	Y
Agreeableness	Theft\P	-0.848	0.002	-0.8533	0.0017	Y
Agreeableness	Help\P	0.759	0.011	0.7289	0.0168	Y
Honesty-Humility	Help\P	0.051	0.89	0.0193	0.9577	NSI

Appendix. D Power-Stability Data Schema

Data	Description
Stability percentage	Total proportion of rounds where a hierarchy has been maintained
Power distribution type map	Total stability percentage, average strength, number of times for each agent involved, and frequency for each of the types of power distribution (Democracy, Dictatorship …) For example if, in a game of 20 rounds played by agents A, B, C, D and E, the following distributions are found: Democracy of strength 0.9 between rounds 1 and 5 and Ruling-Class of strength 0.7 between rounds 6 and 16 of agents A, B, C, this map will be: Democracy → (0.2, 0.9, [A → 1, …, E→1], 1), Ruling Class → (0.5, 0.7, [A→1, B→1, C→1], Slavery → (0, 0, [], 0), …

Appendix. E Social Analysis Result Data Schema

Key	Data	Type
SocialMetrics	Social-Metrics Data Object	General analysis
PowerStability	Power-Stability Data Object	
WealthStability	Stability percentage: Proportion of rounds where wealth hierarchy has been maintained	
AgentToHierarchyPosition	A map of agents to its position in the most stable power hierarchy	
SocialStructures	A list of all of the found stable social structures	
AgentToGroupCount	A map of agents to the number of social structures they can be found in	
SocialGroupMetrics	Social Metrics Object A SocialMetrics object will be generated to analyse each social structure individually and will be merged into a single one by averaging the corresponding data points.	Analysis within social structures.
GroupPowerStability	Power-Stability Data Object Generated through similar method as SocialMetrics Object.	
CompetingGroupMetrics	Social-Metrics Data Object	Analysis regarding to competition between identified social structures. The social groups are treated as a single entity in the exchange network
CompetingPowerStability	Power Stability Data Object	

Appendix. F Full list of assumptions

#	Area	Assumptions
1	Appraisal accuracy	An agent's accuracy in appraisal of other agents' competency and personality increases with: The total number of interactions between them Appraisal skill
2	Competency improvement	Higher the conscientiousness (C) of an agent, the quicker its mining competency improves Higher the extraversion (E) of an agent, the quicker its appraisal competency improves
3	Imprisonment	The higher the number of times an agent steals, the less likely it is for it be caught. Agents lower in conscientiousness and high in emotionality are more likely to be caught
4	Friendship	The level of friendship between two agents: Increases as (and in order of importance): Number of theft interactions between them decreases Number of friendship interactions between them increases Number of help interactions between them increases Similarity in personality increases Number of mentorship interactions between them increases Number of help interactions increases Number of caught theft interactions decreases The higher the level of forgiveness (A1) an agent has, the lower the effect of theft interactions have in decreasing friendship Friends: Agents with a high level of friendship between them
5	Stealing aversion	Agents high in agreeableness (A) are less inclined to steal Agents high in honesty-humility (H) are less inclined to steal Agents are less inclined to steal from **friends**
6	Risk aversion	Increases as Emotionality (E) increases Curiosity (O2) decreases Unconventionality (O4) decreases
7	Help	Agents are less likely to help other agents with a higher amount of gold earned. Agents' willingness to help is proportional to their level of **stealing aversion**.
8	Theft cost	Increases each time an agent is caught stealing within the same round Increases with **imprisonment** level

9	Theft	Perceived cost: Increases with **risk aversion** and with **theft cost** Potential gain (or greed): Increases as the positive difference between the victim and this agents wealth increases Agents are more likely to steal when: Their **stealing aversion** is lower Potential gain outweighs perceived cost
10	Interaction	Agents high in extraversion and openness to experience are more likely to interact with other people
11	Teachable	Agents are more teachable when: High in curiosity (O2) High In self-awareness (O3) High in conscientiousness (C) Low in anxiety (E2)
12	Teaching	Agents are better mentors when: High in flexibility (A3) High in patience (A4) High in fairness (H2) High in sincerity (H1) Highly **teachable**
13	Willingness to be mentored	Perceived need: Increases as the skill-difference between the agent and its mentor increases. Perceived cost: Increases as the potential cost of the mentorship increases. Agents are more likely to be mentored if … Agent is highly **teachable** Mentor is a good at **teaching** Mentor is their **friend** The perceived need for a mentor outweighs the cost of mentorship
14	Willingness to mentor	Agents are more likely to mentor if … Agent is highly **teachable** Mentee is their **friend**

Appendix. G Complete Mathematical Model

Assumption	Formula
1	$accuracy_{percentage} = appraisal_{skill} \times \frac{n}{3a}$ $n = total\ number\ of\ interactions$ $a = average\ number\ of\ interactions\ in\ the\ first\ round$
2	$Hexaco_P(D) = average_{score}(D), \quad 0 \leq Hexaco_P(D) \leq 1$ $D \in \mathbb{P}([H, E, X, A, C, O] \cup Hexaco_{facets})$ $mining_{skill} = mining_{skill} \times (1 + 0.01 \times Hexaco_P(C))$ $appraisal_{skill} = appraisal_{skill} \times (1 + 0.01 \times Hexaco_P(X))$ $C = conscientiousness, X = extraversion$
3	$caught_{probability}(x: Agent) = 0.9 - \frac{0.4n_x\left(1 + Hexaco_P(C_x)\right)}{a(1 + Hexaco_P(E_x))}$ $n_x = number\ of\ thefts,\ E_x = emotionality, C_x = conscientiousness$
4	$f(x, y) = computes\ how\ friendly\ agent\ y\ is\ to\ agent\ x$ $A = number\ of\ friend\ interactions\ between\ x\ and\ y$ $B = HexacoPersonalitySimilarity(x, y)$ $C = number\ of\ help\ interactions\ between\ x\ and\ y$ $D = number\ of\ theft\ interactions\ between\ x\ and\ y$ $E = number\ of\ caught\ thefts$ $F = number\ of\ mentorship\ interactions\ between\ x\ and\ y$ $G = total\ number\ of\ help\ interactions$ $\boldsymbol{Variable\ Importance\ in\ f(x, y)}: D > A > C > B \geq F > G \geq E$ $n = number\ of\ interaction\ rounds$ $p = number\ of\ agents$ $f = Hexaco_P(A1_x) \quad ,where\ A1_x = Agreeableness: forgiveness$ $f(x, y) = 0.5 + \frac{1}{40}\left(2B + \frac{1}{n}\left(10A + 5C + 2F + \frac{G}{p}\right) - (1 - 0.9f)(\frac{17D}{n} + \frac{E}{p \times n})\right)$ $0 \leq f(x, y) \leq 1,\ \ f(x, y) > 0.5 \rightarrow Friends$
5	$stealing_{aversion}(x, y) = aversion\ of\ agent\ x\ to\ stealing\ from\ y$ $av(x) = \frac{1}{2}(Hexaco_p(A_x) + Hexaco_p(H_x))$ $A_x = Agreeableness\ (agent\ x), H_x = Honesty\ Humility\ (agent\ x)$ $stealing_{aversion}(x, y) = av(x)^{1-4\times av(x)(friend(x,y)-0.5)}$ $0 \leq stealing_{aversion}(x) \leq 1$
6	$risk_{aversion}(x: Agent) = \frac{1}{2}\left(1 - Hexaco_p(O2_x, O4_x)\right) + Hexaco_p(E_x)$ $O2 = Inquisitiveness, O4 = Unconventionality, E = Emotionality$ $0 \leq risk_{aversion}(x) \leq 1$

7	$help(x,y) = probability\ of\ agent\ x\ helping\ agent\ y$ $help(x,y) = \frac{wealth(x)}{2 \times wealth(y)} \times stealing_{aversion}(x,y)$ $0 \le help(x,y) \le 1$
8	$theft_{cost}(x) = \frac{1}{3}(risk_{aversion}(x) + 2\big(1 - caught_{probability}(x)\big)^{n+1})$ $n = number\ of\ times\ caught\ this\ round$
9	$theft(x,y) = probability\ of\ agent\ x\ stealing\ from\ agent\ y$ $theft(x,y) = \big(1 - stealing_{aversion}(x,y)\big) \times theft_{cost}(x) \times \left(\frac{wealth(y)}{wealth(x)}\right)$ $0 \le theft(x,y) \le 1$
10	$interact(x) = probability\ agent\ x\ will\ interact$ $interact(x) = \frac{1}{4}(Hexaco_p(O_x) + 3 \times Hexaco_p(X_x))$ $O_x = Openness, X_x = extraversion$
11	$Teachable(x) = \frac{1}{2}(Hexaco_p(O2, O3, H4, C) - Hexaco_p(E2))$ $O2 = inquisitiveness, O3 = creativity, H4 = modesty, E2 = anxiety$
12	$mentor(x) = how\ desirable\ agent\ x\ is\ as\ a\ mentor$ $mentor(x) = \frac{1}{2}(Hexaco_p(A3, A4, H2, H1) + teachable(x))$ $A3 = flexibility, A4 = patience, H2 = fairness, H1 = sincerity$
13	$cost(x) = 1 - \frac{mentorship_{cost}}{wealth(x)}$ $mentor_{want}(x,y) = \frac{1}{2}\big(mentor(y) + friend(x,y)\big)$ $accept_{mentor}(x,y) = competency_{difference}(x,y) \times cost(x) \times mentor_{want}(x,y)$ $0 \le accept_{mentor}(x,y) \le 1$
14	$accept_{mentee}(x,y) = \frac{1}{3}(teachable(x,y) + mentor(y) + friend(x,y)$ $0 \le accept_{mentee}(x,y) \le 1$

Appendix. H Archive File

Folders

- **Code**
    - **Agent.py** – Class that defines StaticAgent behaviour
    - **LearningAgent**.py – Class extends Agent.py with q-learning mechanism
    - **Experiment**.py – Class handles drawing graphs, running assumption checking experiment, running the main experiment, sets up training files, resumes training from config files, plots 2D and 3D graphs to visualise the assumption checking data, processes main experiment results, and sets up simulation
    - **Analysis.py** – Uploads simulation data to database, runs queries to receive necessary data for assumption checking experiment
    - **ServiceGUI.py** – Processes requests to GUI
    - **Helper.py** – Helper functions
    - **GoldMiningEnvironment.py** – Code for the environment
    - **SocialGroupGUI.py** – Uses graphics.py to display the simulation
    - **SocialAnalysis.py** – Social phenomena detection software
- **Test**
    - **TestingSocialAnalysis.py** – Testing suite to test all aspects of SocialAnalysis.py
    - **Testing.py** – Testing suite to test that the simulation is working
- **Experiment**
    - **Run 0 Agent Analysis Data.csv**
    - **Run 0 Power Stability.csv**
    - **Run 0 Social Structure Data.csv**
    - **Run 1 Agent Analysis Data.csv**
    - **Run 1 Power Stability.csv**
    - **Run 1 Social Structure Data.csv**
    - **Run 2 Agent Analysis Data.csv**
    - **Run 2 Power Stability.csv**
    - **Run 2 Social Structure Data.csv**

Appendix. I Original Project Brief

Student Name:

Kimathi Nyota (kn3g17)

Supervisor Name:

Professor Adam Prugel-Bennett

Title:

Can reinforcement learning and genetic algorithms be applied to an agent in a simulated social environment to trigger a 'personality' change with the purpose of improving its social status.

Project description:

I'll be simulating people as agents by assigning certain personality points such as assertiveness, conscientiousness, agreeableness and competence. The agents will exist in a simulated social environment consisting of a set of different types of agent-to-agent interactions along with rules to govern these interactions that considers the agents' personality. Each agent interaction will either increase or decrease the agent's place in the defined social hierarchy of the environment. The goal of the agent will be to ascend to the top of the social hierarchy.

I'll be applying reinforcement learning and genetic algorithms to these agents so that they can make gradual changes to their own personality in order to improve their social status.

The goals of the project will be to accurately simulate people and their interactions in a social environment and to investigate whether these agents can learn to improve their status via machine learning.

Scope: Below are the key project milestones and considerations

1. Identify defining personality traits within people that can be assigned to agents
   Produce a computational model of people as agents
2. Identify how people form social hierarchies in different social environments.
   Produce a computational model of a generalised social hierarchy
3. Identifying general types of interactions that people may have within any social environment.
   For each interaction: determine how the personality of the people affects these interactions and determine how the social hierarchy will change.
   Produce an algorithm that uses agents, interaction type and current social hierarchy as inputs and returns an updated social hierarchy.
4. Only apply reinforcement learning and genetic algorithms to a single agent interacting with a single static agent of a higher social status.
   Determine whether it is possible for the agent to increase its social status.
   Note: Static in this context refers to a non-learning agent with a fixed 'personality'
5. Increase the difficulty by using more static agents (leading to more interactions).
6. Increase the difficulty further by using more non-static agents
7. Increase the difficulty further by increasing the complexity of the social environment model e.g. by incorporating alliances and favouritism