Gift Giving Between Computational Agents in the Repeated Prisoner's Dilemma Game

by

Amy Corbett

An essay submitted to the Department of Economics
in Partial fulfillment of the requirements for
the degree of Master of Arts

Queen's University
Kingston, Ontario, Canada

July 2008

Abstract

In this paper an evolutionary process is simulated in order to view the effects of gift giving between computational agents in a prisoner's dilemma game. Gift giving has evolved to be a stable social custom in many societies, and gifts are exchanged between partners in many relationships. Particularly at the beginning of a relationship, gifts can be used to signal intentions or a partner's type. Gifts themselves may have low value to the receiver, but the giving of gifts has value in and of itself, and can lead to cooperation in the prisoner's dilemma game.

Acknowledgements

I would like to thank Prof. Carmichael for his excellent suggestions, advice and guidance. I would also like to thank Michael for his support and my mother and John for their support and supplying my laptop.

Contents

List of Figures

List of Tables

1 Introduction

In many societies today, people are not restricted to the same match or partner for all of eternity. They are allowed to look for new business partners or new romantic relationships. In the standard repeated prisoner's dilemma game, partners are locked into a match for either a set number of times or forever. Cooperation can be achieved through the threat of punishment if the probability that the relationship will continue is high enough. However, given that in much of real life people can leave partnerships, Carmichael and MacLeod (1997) developed a model in which there is a cost at the beginning of a relationship. In their model, cooperation can be achieved even without the threat of punishment. This paper simulates computational agents to observe the dynamics and to test the assumptions implied by Carmichael and MacLeod's model. When costs, taking the form of gifts, are imposed at the beginning of a relationship cooperation can be achieved.

Komter and Vollebergh (1997) suggest that there are four kinds of gifts: presents, food (having guests over for dinner), lodging (offering guests a place to stay in one's house), and care or help. Gift-giving, or costs at the beginning of relationship, can be found throughout society. For example, when couples begin dating or courting they often give gifts of flowers or candy. In the model used in this paper the type of gift is important. It has to cost more to buy then it is worth to the receiver, thereby imposing a cost when implemented in a symmetric prisoner's dilemma game. As well, the gift must depreciate in value quickly. This prevents a parasite from recycling the gift. A parasite in the model is a strategy defined by an agent who gives a gift, defects and then searches for a new match. If the parasite could re-use the gift that was given to them, then they would only have to buy the gift once, which reduces the relative value to cooperation. In the model cooperation can be achieved with the clubbies. Clubbies are agents using the club strategy, which is defined by giving a gift, and if they receive a gift greater than or equal to a set gift bound, they will cooperate.

The gifts that are given, and the giving of gifts itself, seem to have evolved in society to being a stable social custom. A social custom is defined as a behavioural rule or strategy passed down from parents to their children. Many children follow this custom unquestioningly. In the model, this drives the dynamics and is introduced as an evolutionary step. While much of the next generation simply copies what their parents did, many copy the strategy that performed the best on average in the previous generation. If the parasites did better than the clubbies, for example, many of the new generation will copy this strategy.

This paper is organized as follows: in the next section a review of gift-giving in the literature is given. In section three an outline of the model is presented. Section four presents the results of the simulations and robustness tests. Section 5 offers a conclusion.

2 Gift-giving in the literature

When van de Ven (2000) discusses gift-giving in the literature, he argues that there are two important aspects that must be considered: reciprocity and adequacy/efficiency. As well, there are five main motivations for gift-giving in the literature: altruistic gift-giving, egoistic gift-giving, strategical gift-giving, and gifts given to achieve fairness or for survival. In addition, gifts serve both social and economic functions. In this section, reciprocity and inefficiency are first discussed, followed by a discussion of the different motivations for gift-giving used in the literature today.

2.1 Reciprocity and efficiency

Although the giving of gifts may seem to be voluntary, many gifts have strong reciprocal properties. For example, consider blood donations. At first glance it would appear that the giving of blood would be a gift to an unknown that is not reciprocated. However, Arrow (1972) suggests that "one gives good things, such as blood, in exchange for a generalized obligation on the part of fellowmen to help in other circumstances if needed." Thus, even though a gift may not be immediately reciprocated, the giver expects some return down the line. Arrow also emphasises the “implicit nature of the social contract”. Many gifts act as a type of contract obligating the receiver to reciprocate.

Bellemare and Shearer (2007) found evidence of reciprocity from a field experiment. In their experiment, an unexpected gift is given to workers in a tree planting firm. After controlling for the weather and other shocks, they find that workers increased their productivity on the day that they received the gift. Bellemare and Shearer also find that reciprocity and tenure are positively related. They suggest that “repeated interaction associated with longer tenure may reduce social distance between the workers and the manager, leading to stronger reciprocal behaviour”.

Akerlof (1982) suggests that gifts between employees and employers are given partly because workers care about each other. In his paper, instead of working the minimal required level workers work up to a norm that is determined by the employees, and this is a gift to the employer. In turn, the employer gives back in the form of higher wages, or in the form of a future promotion and the promise of higher wages. According to Akerlof, in most cases the gift given is approximately in the range of what the recipient expects and he reciprocates in kind.

In Sherry (1983), a discussion of gift-giving in anthropology is offered. According to Sherry, “to avoid feeling inferior and to safe-guard reputation, the recipient must reciprocate.” However, “gifts to individuals perceived as status subordinates - such as the news carrier, the postman, or the waitress - generally carry no expectation of equivalent return”. Thus, people’s status may exempt them from having to give a gift at a certain point in their life or forever. For example, having status of student or transient may exempt one from having to give a gift. Youth or people who have hit a rough patch may not be expected to reciprocate in a balanced fashion.

Thus, it appears as if gifts within certain relationships are reciprocated, and others are not. It is important in discussing gift-giving to explain when and why gifts are reciprocated. In this paper agents have an obligation to reciprocate. Those following the club strategy have an especially strong duty to reciprocate as they use gift-giving at the beginning of relationships to create a long-term bond.

The inadequacy or inefficiency of gifts was also highlighted by van de Ven (2000) as being an important consideration when analyzing the giving of gifts. Although standard microeconomics tells us that cash is never worse than a gift of the same value, society seems to have evolved so that in many relationships one cannot give cash, as it would seem rude or inappropriate. For example, it is nice to bring your date flowers, but it would be inappropriate to give them the cash equivalent. Birthday gifts and business gifts are often inadequate. One possible reason for the inadequacy or seeming inefficiency of gifts could be that symbolic utility is neglected, and that if this symbolic utility was taken into consideration then gifts could be made more adequate/efficient.

Camerer (1988) suggests several reasons for why inefficient gifts are given. Two of these are that inefficiency is based on convention and is arbitrary and that gifts are signals about how much a gift giver knows about the receiver's tastes and some inefficiency will occur when a giver guesses wrongly.

According to Komter and Vollebergh (1997), "although many gifts are transformable into economic transactions (care or help can be bought and sold, presents can be stripped of any personal meaning and become merely economic - for example, coupons or money gifts), many people, at least when they have certain material resources and a certain amount of time at their disposal, seem to prefer the personalized form of gift giving - giving as a means to express personal feelings toward other people - above the economic form". This suggests that gifts have value beyond their economic value, and may be an expression of feelings, which is why they often appear to be inefficient or inadequate to an outside observer of the gift exchange. This point is also emphasized by Harsanyi (1969) in a discussion of "irrational" behavior he says that "people often derive considerable psychological satisfaction from symbolic actions".

In the remainder of this section, the five main motivations for gift-giving are discussed.

2.2 Altruist gift-giving

In economics, altruism is often described by an agent with a utility function that contains other's utility. Optimizing agents will reach an equilibrium between personal consumption and the consumption of those who enter into their utility function. This theory does not explain why some gifts are inefficient. However, it does explain why some gifts may not be reciprocated, and could help explain why anonymous gifts may be given, such as charitable donations.

Parry (1986) provides a discussion of gifts in different civilizations. He suggests that even though altruistic gifts do not seem to have reciprocal properties, in some cultures, "the gift does indeed return to the donor, but it does so as the fruits of *karma*". If one believes in karma, even if their utility function does not contain other's utility, they may benefit from giving a seemingly altruistic gift.

According to Akerlof (1982), "if workers have an interest in the welfare of their coworkers, they gain utility if the firm relaxes pressure on the workers if hard pressed; in return for reducing such pressure, better workers are often willing to work harder." This suggests that workers utility may contain the utility of their co-workers, and that by helping their co-workers they gain some happiness in return.

In Arrow's (1972) discussion of Titmuss, "the giving of blood is giving not to specific individuals but to an anonymous recipient. The motives for such giving are regarded as more definitely altruistic than those for giving to individuals". He also infers from Titmuss that "the motives for giving blood can be divided into three types: a generalized desire to benefit others, a feeling of social obligation, and a response to personal social pressures, as in the case of donations to known recipients or responses to institutional blood drives."

Even the act of donating blood may not be purely altruistic. All gifts seem to lie somewhere on a spectrum in between pure altruism and pure selfishness. Harsanyi suggests that "people's behavior can largely be explained in terms of two dominant interests: economic gain and social acceptance." This idea will be further discussed in the next subsection on egoistic gift-giving.

2.3 Egoist gift-giving

Egoist gift-giving is similar to altruist gift-giving in that the giving of gifts benefits the giver in both case. However, with altruism the utility of the receiver of the gift enters into the utility function of the giver, and with egoism it does not. van de Ven (2000) suggests that egoistic gift-giving can be further broken down into two classes: gifts as an exchange mechanism and gifts as a means to obtain social approval.

Gifts as an exchange mechanism are commonly seen in primitive and early societies, especially before the discovery of money or a system of market exchange. Kranton (1996) develops a model in which some agents engage in reciprocal gift exchange and the rest engage in a monetary market exchange (where money replaces gifts as the exchange mechanism). She suggests that there is an interaction between the two means of exchange. With monetary exchange, agents have access to a variety of goods, yet they have to search for trading partners. With reciprocal gift exchange agents do not have to search as much for partners, yet they are limited only to the commodity that their partner produces. Thus, if it's difficult to find partners gift exchange may be a better means of trade. However, Kranton emphasizes that if too many people are initially involved in reciprocal exchange, the economy could be stuck at a socially inefficient equilibrium. As well, the presence of a market economy reduces the punishment if an agent defects by not reciprocating a gift. However, according to Kranton, "repeated interaction, reputation mechanisms, and long-term agreements, ... can mitigate the gains from ex post opportunism." Kranton's model shows how a reciprocal exchange system could operate and discusses the benefits to exchanging gifts.

The second reason for egoistic gift-giving highlighted was the giving of gifts as a means to achieve social approval. For example, one would give a gift to either impress the receiver or to help create a social bond with the receiver.

Andreoni (1989) develops a model in which altruistic giving is not "pure". He assumes that people get a warm glow from giving. Thus, when giving a seemingly altruistic gift, such as volunteering labour or donating to a charity, people do it for both a selfless reason (to "do their bit"), and an egoistic reason (the warm glow they receive from the social approval). Andreoni's work highlights the emotions involved when people give and receive gifts.

Ruffle (1999) develops a game theoretic model of gift giving where emotions matter. In his model, beliefs directly enter into the payoff functions of the agents, and agent's emotions play a role when there is a difference between the outcome of the game compared to the agent's beliefs prior to the game being played. For example, if an agent expected to receive a small gift but then turned out to receive a large gift, they are surprised, and this positively enters into their payoff. The giver of the gift also takes pride in surprising their partner, which enters positively into their payoff. The gift giver's choice of action depends on the cost of the gift as well as his beliefs about his partner's expectations. Gift-giving, non-gift giving and mixed-strategy equilibrium may emerge depending on the beliefs and the costs of gifts. For example, if the cost of gifts is too high, this will prevent an agent from purchasing a gift for their partner, even though their partner is expecting one. Ruffle's model especially emphasizes expectations associated with giving and receiving gifts, and how expectations change depending on the closeness of relationships.

Ruffle quoting Malinowski (1992) states that "the view that the native can live in a state of individual search for food, or catering for his own household only, in isolation from any interchange of goods, implies a calculating, cold egotism, the possibility of enjoyment by man of utilities for their sake. This view ... ignore[s] the fundamental human impulse to display, to share, to bestow. [It] ignore[s] the deep tendency to create social ties through exchange of gifts." Thus, egoistic gift-giving often takes the form of gifts given to create social ties. By the giving and receiving of gifts, those involved in the exchanges are bonded. This idea of giving a gift to create a bond with the receiver could be used in a strategic way, which is discussed in the next section.

2.4 Strategical gift-giving

Strategical gift-giving refers to the idea that a gift is given with an ulterior motive, beyond giving for the sake of giving. Strategical gift-giving in the literature often takes the form of gifts being given as a signal. This could be to signal a giver's type or intentions. In Carmichael and MacLeod, gifts are given by clubbies to signal their intentions to cooperate. Gifts are also given by parasites to try and fool the clubbies into cooperation. A gift in Carmichael and MacLeod can be viewed as a costly message, and in their model this is necessary to achieve a good, cooperative equilibrium.

Camerer's (1988) model is also a good example of strategical gift-giving. Camerer suggests that gifts serve as signals and that inefficient gifts may be better signals. He develops a two-stage game-theoretic model in which gifts are signals of a person's intentions or expectations about future investment in a relationship. Camerer achieves inefficiency by introducing a cost to play the signalling game. In the two-stage gift-giving game he shows the areas where different equilibria may exist, and discusses where separating and pooling equilibria occur. If the size of the gifts is appropriate a separating equilibrium could occur and a player's type could be revealed by whether or not they give a gift.

Many gifts given in real life have some elements of strategy involved. For example, a child may give a teacher an apple for many plausible reasons, one of these is to try and get a better grade. Gifts given as bribes are commonly heard of from college sports teams trying to recruit a good player to their team.

2.5 Fairness

Another reason or motivation for the giving of gifts may be to achieve an equitable outcome, and so gifts are given to redistribute income. It can be assumed that people have a preference for a fair outcome. People dislike to be richer or poorer and have the possibility to redistribute by giving, in an attempt to obtain a fair, or more equal, outcome. An example of a gift given in fairness could be an anonymous charitable donation.

Fairness plays a role in Camerer's (1988) model. In his model, the size of the gift given depends on the agent's endowment. If the endowments differ enough, the gift-giving could become completely one-sided, therefore the gift imperfectly redistributes income.

Another type of fairness consideration in gift-giving could be that people prefer to exchange gifts of the same size. Ruffle (1999) tries to include a role for fairness by adding a fairness term to player's utility functions. Equal exchanges (i.e. gifts of the same size) "are assumed to contribute positively to players' utility. Unequal exchanges are assumed to contribute negatively to utility: a player may feel cheated if she bears the expense of the costly action while her opponent takes the cheap one and guilty if the situation is reversed." Fairness considerations can be seen throughout societies. For example, many parents try to give their children gifts of the same size or value at Christmas and birthdays.

Rabin (1993) develops a game-theoretic model with a consideration for fairness. In his model, he tries to incorporate the idea "that people are willing to sacrifice their own material well-being to help those who are being kind" and that "people are willing to sacrifice their own material well-being to punish those who are being unkind." Rabin suggests that if somebody is being nice to you, you will in turn be nice to them, and if they are being mean to you, you are in turn mean to them. While Rabin does not explicitly talk of gifts, one can easily see how gifts can be the material sacrifice given to a partner to convey the emotion of niceness or meanness, and to achieve what the agent believes to be a fair outcome.

2.6 Survival

Carmichael and Macleod's model, upon which this paper is based, develops an evolutionary model of gift-giving. In their model, a strategy does well in the population over time if it performs better than average. Those playing the club strategy exchange gifts at the beginning of a relationship, which allows them to signal their intentions to their partner. If they are matched with another club, they can achieve a cooperative outcome and enter into a long-term relationship, which benefits them both. Thus, the gift can allow the strategy to perform better than other strategies, increasing the survival chances of the strategy.

Carmichael and Macleod's model and the idea of survival will be discussed in detail in the Model section.

2.7 Summary

The five motivations for gift-giving discussed (altruist, egoistic, strategical, fairness and survival) can be modelled in different ways, and there is much cross-over between each of the modeling's. As well, there are multiple reasons for every gift that is given. For example, many wedding gifts take the form of cash, and this can be seen as transferring wealth from those at a stage in life who have extra wealth, to those at a stage in live when they need money (fairness, survival). However, many wedding gifts take the form of items which have sentimental value (altruistic, egoistic and strategical). Another example is charitable donations. A gift to a charity can be motivated by altruism, egoism and given to achieve a more fair income distribution.

The gifts used in the model presented in this paper also fit into more than one of the categories discussed, the main two being strategical and survival. There is a strategical element in that the gifts are being used to try and signal an agent's intentions to enter into a long-term cooperative match, and there is a survival component in that a strategy must do well in the simulations to survive, thus gifts may need to be given by agents to be the surviving strategy. The model simulated in this paper will be discussed in more detail in the next section.

3 The Model

It is assumed that there are two types of agents, denoted as i's and j's, and time is discrete. An 'i'-type agent is always matched with a 'j'-type agent, never with another 'i', and vice versa. Agents in the model are randomly and anonymously matched and every agent is matched in every period. In each period agents in a match simultaneously send and receive gifts. The partners then play a prisoner's dilemma game, as shown in Table 1. Agents receive payoffs and make separation decisions. There is a positive probability of death (1-ρ) for each agent in every period and it is assumed that both agents in the match die, and are replaced with new agents. New and separated agents are then randomly matched, and the process repeats. New agents either copy their parent's strategy, copy the strategy that is best performing, or are randomly assigned a new strategy. Separated agents use the same strategy they were assigned. This process of matching, sending a gift, playing the game, making separation decisions and facing the probability of death repeats for a fixed number of periods, and then an evolutionary step occurs. In the evolutionary step all of the agents are replaced, and a new population or generation of agents is formed. A proportion of the agents use the strategy that received the highest average discounted payoff in the previous generation, another proportion copy their parent's strategy, and a third small proportion are randomly assigned a strategy. This new generation then sends gifts, receives payoffs and makes separation decisions, as described above. This process repeats for a fixed number of periods. The flow of the model is presented below (Table 2), and the remainder of this section discusses this process in further detail.

Table 1: Prisoner's Dilemma Game

a^i, a^j	C	D
C	A	0
D	B	1

Table 2: Model Flow

1. Agents match
2. Agents send and receive gifts
3. Matches play the prisoner's dilemma game according to their assigned strategy
4. Agents make separation decisions based on the gift received and the outcome of the game
5. Agents face a probability of death
6. Steps 1-6 repeat for a fixed number of times for separated agents, steps 3-6 repeat for a fixed number of times for matched agents
7. An evolutionary step occurs, where agents are replaced by a new generation
8. Steps 1-7 repeat for a fixed number of times

3.1 Step 1: The matching market

New and separated agents enter into the matching market. In the matching market 'i'-type agents are randomly matched with 'j'-type agents. In the computational environment of this paper, a finite number of agents is used, and therefore there is a positive probability that the agent was previously matched with the same partner. However, it is assumed that agents have no history of prior matches with the agent.

3.2 Step 2: Gifts

The gift-giving technology of this model, which can be interpreted as a costly message, allows agents playing a new strategy to use a 'secret handshake' to identify each other, while appearing normal to other agents. If agents could accurately identify the newcomers, any strategy that always defects against outsiders would never be invaded.

Secret handshakes and message sending allows for cooperation to emerge, and allows for efficiency. In this paper a simplified version of Carmichael and MacLeod's model is used with only two strategies, and it is assumed that all agents send the same costly message.

When gifts are given, all agents give a gift at the beginning of a relationship of the same size. This is a simplifying assumption to ease the computations. In a more complicated model agents could have a choice to send a gift or not, and a choice of what value of gift to purchase and send. The gift itself is assumed to have lower value to the receiver than it costs the sender, and to only benefit the receiver in the period it is sent. Because of these assumptions and because gifts are given by both agents in a new match and payoffs in the game are symmetric, the gift only enters into the payoff function of the sender as the difference between the cost to the giver and the value to the receiver. If a match continues, it is assumed that agents no longer send gifts.

3.3 Step 3: Prisoner's dilemma game

Agents in a match play a standard repeated prisoner's dilemma game (Table 1), in which the payoffs are assumed to satisfy the condition that $2A > B > A > 1$. In the prisoner's dilemma game, each agent can choose to either cooperate (C) of defect (D). If both agents cooperate, then they both receive a payoff of A. However, if instead one agent defects and the other cooperates, then the agent that defected receives a payoff of B, and the agent that cooperated receives a payoff of B, where $B > A$. If both agents defect, they both receive a payoff of 1. If both agents are cooperating, they each have a private incentive to defect and get a payoff of B. But if both agents defect, they are worse off than if each was cooperating. So, while (defect, defect) is the only Nash Equilibrium in the one-shot game, (cooperate, cooperate) would be a Pareto-improvement.

It is assumed that there are two types of agents, clubbies and parasites, where clubbies are following the club strategy and parasites are following the parasite strategy. Agent's following the club strategy give a gift at the beginning of the match. If they in turn receive a gift with value greater than or equal to an acceptable level (g), they will cooperate (C) and if not, they will defect (D). If the appropriate gift was given and their partner cooperated, they will stay in the match.

Agent's following the parasite strategy give a gift, but always defect and quit the match. Carmichael and Macleod (1997) highlight two types of parasites that are likely to invade the population: parasite consumers and parasite recyclers. Parasite consumers give the gift, defect and quit. It is assumed that parasite consumers consume the gift. However, parasite recyclers, instead of consuming the gift, give a gift, defect and quit, but then use the same gift that they received in their next match. If a relationship is long enough, the discounted payoff for clubbies from giving of a gift once and remaining in a match may be higher than the discounted payoff for parasites consumers who have to buy a gift in every period. However, because parasite recyclers only have to buy a gift once, if they are allowed to exist they can take over a population. In the simulations, both parasite consumers and parasite recyclers are tested separately. However, most of the focus is on parasite consumers. This is because, as will be discussed later, society has evolved in such a way as to prevent the recycling of gifts.

3.4 Step 4: Separation decisions

After playing the prisoner's dilemma game, agents decide whether or not to remain in a match or to go back to the matching market to find a new match. Agents base this decision on the message/gift received and on the outcome of the game. If agents receive a gift of value greater than or equal to what is deemed an acceptable gift (g), and their partner cooperated, then they will choose to remain in the match. If the match is already established and has already sent gifts to each other, they do not continue sending gifts, and instead base their separation decisions entirely on the outcome of the game; thus, they will continue in the match if their partner cooperated. Because parasites always defect, only clubs will ever enter into and remain in long-term matches. As well, because all agents give gifts of the same size, as long as this gift is greater than or equal g, all clubs when matched with another club will remain paired with that club until they die or they reach the evolutionary step. This is the key to how a gift-giving custom can emerge: clubbies, if paired with another club only have to give the gift once, whereas parasites have to give a gift in every period. If the parasite has to pay for this gift every time, the benefits to remaining in the match can be higher than the benefits of defection.

When agents separate, either because they or their partner defected or they die, they enter into the matching market to find a new match. In the simulations, there are only two strategies: clubbies and parasites. However, if a new strategy were to enter in, they would go into the matching market. The matching market will consist of mostly defectors, as clubbies and other cooperative strategies will enter into long-term matches. Thus, new strategies entering in need to do well in the matching market if they are to survive. In the evolutionary step, populations are replaced with new generations, and the proportion of agents assigned using different strategies largely depends on how well they did when playing the game.

3.5 Step 5: Death

In every period agents face a positive probability of death. It is assumed that both agents in the match die. This assumption is made to help ease the computation of payoffs. When the match dies, they are replaced with new agents. A proportion of these agents copy their parents strategy, a proportion copy the strategy that is best performing and the rest are randomly assigned strategies. It is assumed that these proportions are the same as those in the evolutionary step. Introducing death in the model can be seen as being most beneficial to the parasite consumers, as it means that long-term club matches may be broken up, and the new agents that replace them have to give a gift. This reduces the average payoff for the clubbies, where it has little effect on the parasite consumers who have to buy a gift in every period anyways.

3.6 Step 6: Repeat steps

Steps 1 to 6 are repeated a fixed number of times for separated agents and steps 3 to 6 are repeated for a fixed number of times for matched agents. The longer the length of this iteration, the more likely it is that clubbies can enter into long-term relationships, and thus the payoff to being a clubby increases.

3.7 Step 7: Evolution

After the generation iterates for a fixed number of times, an evolutionary step occurs. Average discounted payoffs from the iteration are calculated for each strategy. Based on these payoffs, a new generation is formed. These payoffs are calculated by taking the average of all the discounted payoffs for the clubs and the average of all of the discounted payoffs for the parasites. A proportion of the agents copy the strategy of their parents, a proportion copy the strategy with the highest average discounted payoff, and a small proportion are randomly assigned a strategy[1]. This implementation of an evolutionary step is very similar to how populations evolve with evolutionary algorithms that use learning techniques inspired by biology. A good discussion of evolutionary algorithms applied to the iterated prisoner's dilemma game can be found in Van Bragt et. al (2001).

3.8 Equilibrium

If gifts are high enough to deter defection, in the equilibrium all clubs that get matched with other clubs will be cooperating and entering into long term matches. This will drive out the parasites. Because of the probability of death and because a small proportion of agents in each generation is randomly assigned strategies, some parasites

[1] Carmichael and MacLeod do not specifically model the evolutionary step in their paper. However, the selection dynamics described here are discussed in their text.

may still exist in the society. In fact, if there is only one parasite in the economy, so that all of the clubbies are matched and in long-term relationships, this last parasite will be matched with a clubby in every iteration. Because of the assumption that players do not remember the history of matches, the clubby and the parasite will always exchange gifts and the clubby will cooperate every period and the parasite will defect every period. However, because the parasite consumers have to buy a gift in every period, if the game is played enough times and there are enough matched clubbies, the parasite's average discounted payoff will still be lower than that of the clubbies.

The equilibrium concept that is referred to throughout the text is that of neutral stability. An equilibrium is neutrally stable if it is a Nash Equilibrium, and if a small 'mutant' population enters, then the members of the 'mutant' population do no better than the members in the original population.

Carmichael and MacLeod derive a simple equilibrium condition. In their proposition 7, they state that, if the payoffs satisfy

$$\frac{B-A}{B-1} < \rho < 1$$

then a neutrally stable population exists, all with strategies that have the following characteristics: 1)Individuals exchange gifts at the beginning of the match, stay matched until death, and cooperate every period, and 2) All agents send the same-sized gift that is bounded as follows:

$$\frac{A-1}{1-\rho} > g > \frac{B-A}{\rho}$$

.

Because the model is simulated in discrete time and Carmichael and MacLeod's proposition is in continuous time, one must keep in mind that if the model is not simulated for long enough times in between evolutionary steps, the payoffs may not approach close enough the continuous time version and an equilibrium may not exist.

4 Results

The results presented in the following section are results of computer simulations from programs created by the author using StataSE 10[2]. All results presented in this section are sensitive to seeds used by Stata's random number generator. However, multiple runs of each simulation are performed to ensure that the seed is not effecting the main results.

The figures in this section all show percentage vs. time. For example, for graphs showing cooperation, at time 1, this shows the percentage of agents cooperating in the first period of the first generation; at time 2, this is the percentage of agents cooperating in the first period of the second generation; at time 3, this is the percentage of agents cooperating in the first period of the third generation, and so on. The graphs showing payoffs are shown similarly. For some of the payoff graphs there are two lines. The blue line represents the average discounted payoff for the 'i'-type agents, and the red line represents the average discounted payoff for the 'j'-type agents. When there is only one line a representative agent is chosen to avoid confusion in the graphs.

As discussed previously, the equilibrium concept used is that of neutral stability, and the results may be effected by the number of simulations between evolutionary steps. Because of this, it will be concluded that an equilibrium is reached if the fluctuations over time are less than the proportion of agents that can be randomly assigned in every period (this is approximately 5%), and the graphs appear to fluctuate around a constant.

4.1 No messages or gifts

The simulations are begun with a baseline calibration (see Table 2). In this baseline calibration it is assumed that all agents following the parasite strategy are parasite consumers, and there is no gift-giving or message sending (or, equivalently, agents send a gift of 0 and $g > 0$). The simulation is run with 100 agents for 30 evolutions,

[2]A sample of the programs created by the author can be found in the appendix.

each evolution consisting of 10 iterations of the matching phase, gift giving/message sending phase, playing of the prisoner's dilemma game, receiving payoffs, making separation decisions and facing the probability of death. After these 10 iterations there is an evolutionary step, in which the agents are replaced with new agents, some of which copy their parent's strategy (40%), some of them copy the strategy which received the highest average payoff from the previous evolution (55%), and some of whom are randomly assigned a strategy (5%). The initial population starts out consisting of 50% clubbies and 50% parasites. The payoffs in the prisoner's dilemma game are A=10 and B=19. The discount rate is 0.75 and the probability of death (1-ρ) is 25%.

Table 3: Baseline calibration with no gifts

A	10
B	19
ρ	0.75
g	n.a.
% copying parent's strat.	40%
% using highest payoff strat.	45%
% randomly assigned strat.	5%
initial % of clubbies	50%
initial % of parasites	50%

Because there is no message-sending or gift-giving, agents have no way to identify the strategy of the other agents. Thus, one would expected that every agent defects in every period, which is in fact the case (see Figures 1 and 2). The proportion of agents following each strategy may or may not approach a stable distribution (see Figures 3 and 4). Because the payoff from each strategy is the same, new agents entering into a population and the population formed in the evolutionary step, will consist of 60% being randomly assigned a strategy. The average discounted payoff for each strategy is: $1 + 1\rho + 1\rho^2 + ...1\rho^9 = 3.77$.

Both theory and the simulations predict that without message-sending or gift-giving and only club and parasite strategies available to the agents, the outcome of the game will be the same as the one-shot game Nash Equilibrium (defect, defect).

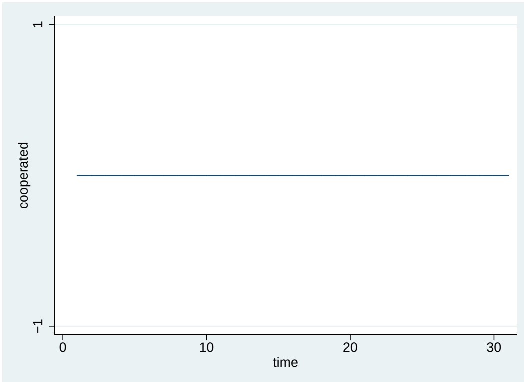

Figure 1: Percentage of agents cooperating (gift 0).

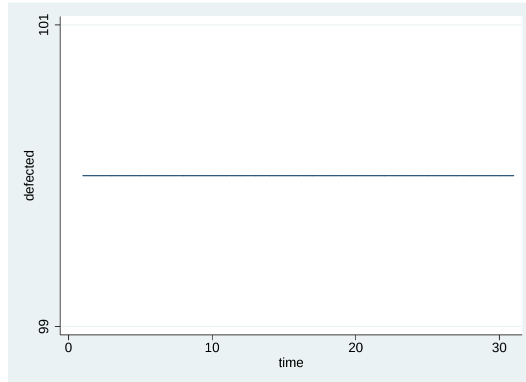

Figure 2: Percentage of agents defecting (gift 0).

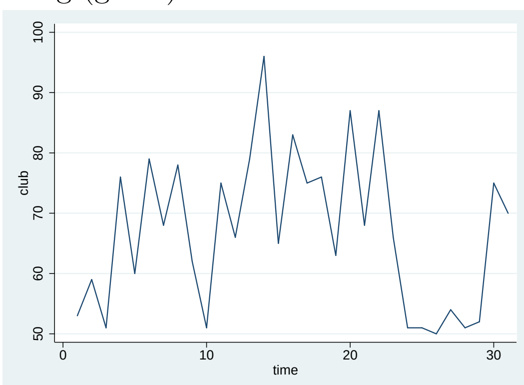

Figure 3: Percentage of agents using the club strategy (gift 0).

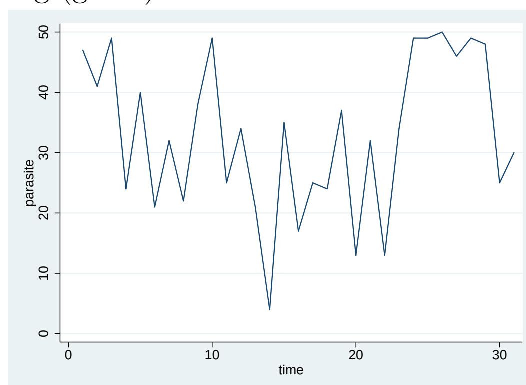

Figure 4: Percentage of agents using the parasite strategy (gift 0).

4.2 Simulations with gifts

Inferentially or implicitly attached strings are a connotative aspect of the gift, social bonds thereby forged and reciprocation encouraged. -Sherry (1983)

In the following, agents are required to send gifts. It can be assumed that the gift itself is a costly message. In all of the simulations every agent sends the same gift/message, or equivalently, every agent sends a gift/message of equal value. This assumption means that agents cannot distinguish between parasites and clubbies based on the gift alone. Thus, in order for gifts to elicit cooperation, the gift itself has to be large enough that the benefit of cooperation and staying in a long-term match is greater than the benefit of giving a gift and defecting. In a more complicated version of the model agents could also be allowed to send costless messages in additions to the gifts, thus acting as a secret handshake allowing the agents to identify other agents of the same type[3].

Because it is assumed that all agents send gifts, in order to tell whether or not a gift would actually evolve in a society, the payoffs from the two strategies are compared with the payoffs in a society with no gift-giving, like that of the previous section.

It is outside the scope of this paper to look for an optimal gift size. In order to do that one would want to simulate between evolutionary steps for an infinitely long period of time.

4.2.1 Baseline calibration with gift-giving

Agents are now required to give a gift of size 13 at the beginning of every new match. If agents choose to remain in a match, they no longer have to give a gift. The calibration of the parameters of the model is shown in Table 3, and besides the gift, is the same as the calibration of the model previously when no gifts were given.

[3]This is actually where the name of the club strategy comes from. Carmichael and MacLeod (1997) first introduce the model allowing for costless messages to be sent; the clubbies send a secret handshake-like message to one another and upon receiving this message they cooperate. The model presented in this paper has this technology, but the main focus is on the gift-giving. Thus, it can be assumed that no agents send costless message, or equivalently, every agent sends the same costless message.

The number of agents cooperating at the start of each generation is shown in Figures 5 and 7. It can be seen in the figures that a stable distribution occurs relatively quickly.

In order to determine if the speed of the results is due to the fact that the initial population consists of 50% clubs and 50% parasites, alongside these figures results are shown for the same calibration as in Table 3, but starting from 10% clubs and 90% parasites. This parameterization also converges relatively quickly (Figures 6 and 8). This shows that one can start from a fairly low amount of clubs in the population and still achieve convergence to a stable gift-giving custom.

Table 4: Baseline calibration with gifts

A	10
B	19
ρ	0.75
g	13
% copying parent's strat.	40%
% using highest payoff strat.	55%
% randomly assigned strat.	5%
initial % of clubbies	50%
initial % of parasites	50%

Next, the average discounted payoffs per generation are shown for both the calibrations with gift-giving (i.e. starting from 50% and 10% clubs) (see Figures 9, 10, 11 and 12). These payoffs are calculated, as discussed previously, by taking the discounted sum of payoffs for each club (parasite), and then averaging them over all the clubs (parasites) in the population at the time. This gives an average payoff value in that generation for each of the strategy types. As expected, based on the similarity of the previous graphs for these calibration, the payoffs are very similar for both calibrations, and for both the 'i'-types and 'j'-types. In order to ease computation, if there are no parasites or clubs in a given population, it is assumed that they have an average payoff of zero. Because of this, the graphs for the parasites appear to have

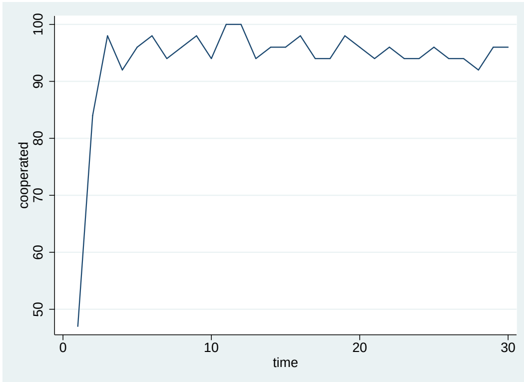

Figure 5: Percentage of agents cooperating, 50% clubbies to start (gift 13).

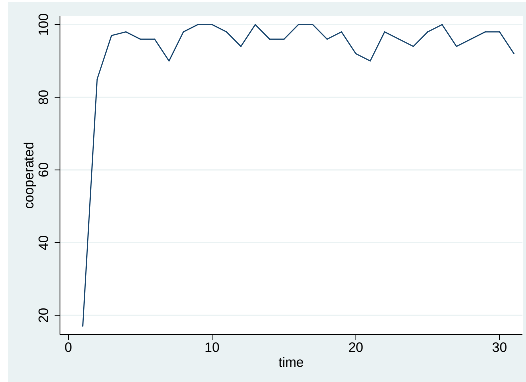

Figure 6: Percentage of agents cooperating, 10% clubbies to start (gift 13).

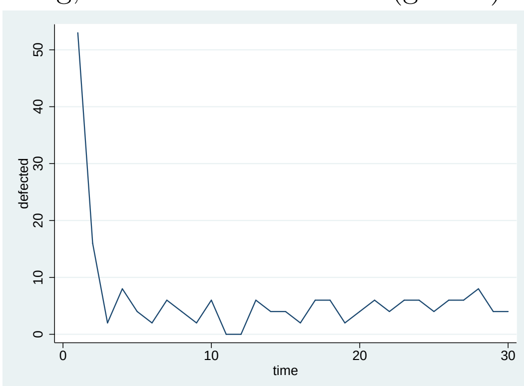

Figure 7: Percentage of agents defecting, 50% clubbies to start (gift 13).

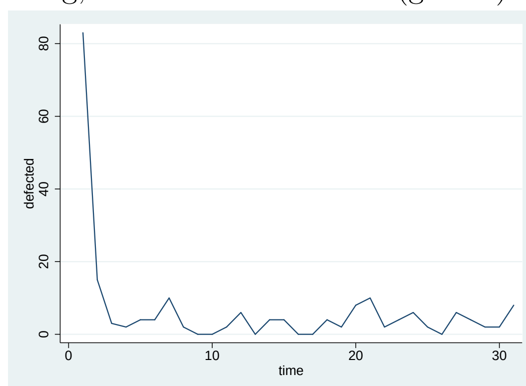

Figure 8: Percentage of agents defecting, 10% clubbies to start (gift 13).

more fluctuation than what is actually occurring[4]. One can also see that the average payoffs for those playing the club strategy is much higher than the average payoff achieved for any strategy when there is no gift-giving (23.78 vs. 3.77). This result may lead one to conclude that gift-giving customs can be profitable for cooperating strategy types. This also lends further credibility to why gift-giving has emerged as a common and stable custom in many societies.

[4]This method of setting the payoff to zero if there is no agent using that strategy is used throughout the simulations.

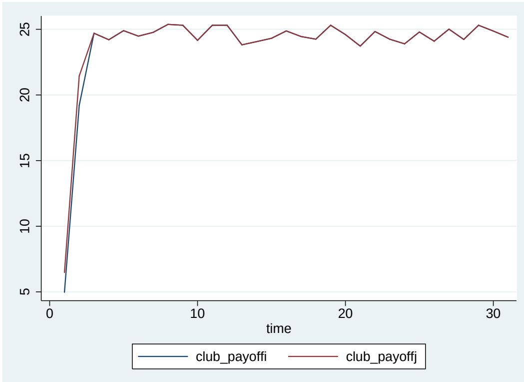

Figure 9: Average club payoff (gift 13), 50% clubbies to start.

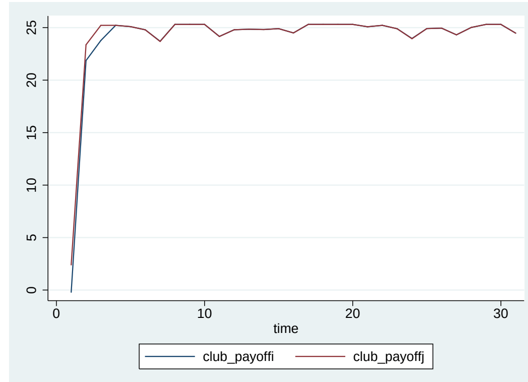

Figure 10: Average club payoff (gift 13), 10% clubbies to start.

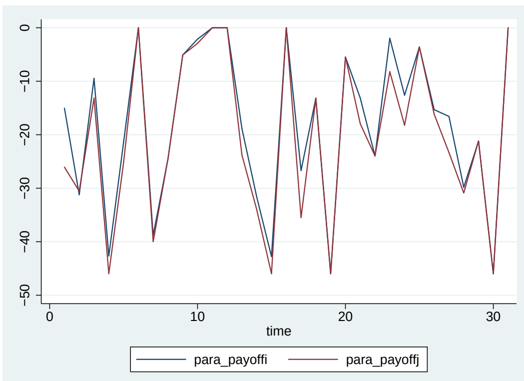

Figure 11: Average parasite payoff (gift 13), 50% clubbies to start.

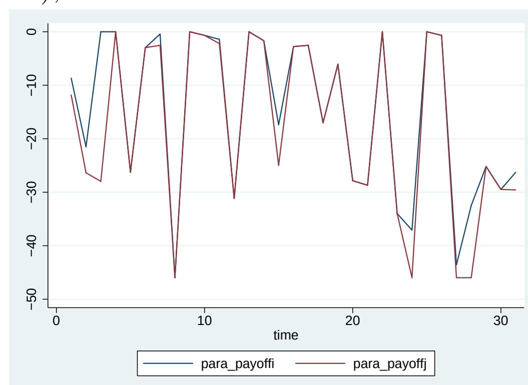

Figure 12: Average parasite payoff (gift 13), 10% clubbies to start.

4.2.2 Simulation check

To further check the calibration, and to test the simulations, result are next presented with the baseline calibration from the previous section starting with 50% clubbies, but now making 100% of new agents copy the highest average discounted payoff from the previous generation. If 100% of the agents copied the strategy that achieved the highest discounted payoffs, the society very quickly converges to 100% clubbies always cooperating (Figure 13), and the payoffs for these clubs approach the maximum that they can attain (they do not reach the maximum due to positive probability of death, as the new agents have to purchase and send a gift) (Figure 14). This maximum value is: $10 - 13 + 10\rho + 10\rho^2 + ... + 10\rho^9 = 33.75$, and the average discounted payoffs that the agents receive is: 25.31.

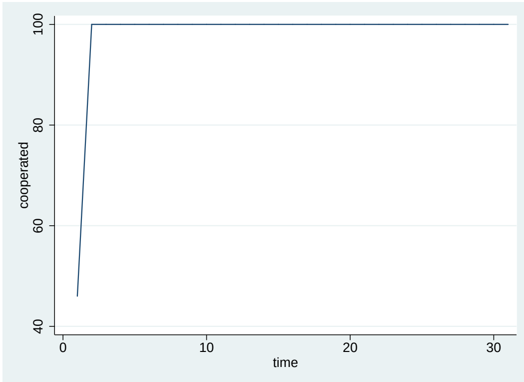

Figure 13: Percentage of agents cooperating (gift 13), simulation check.

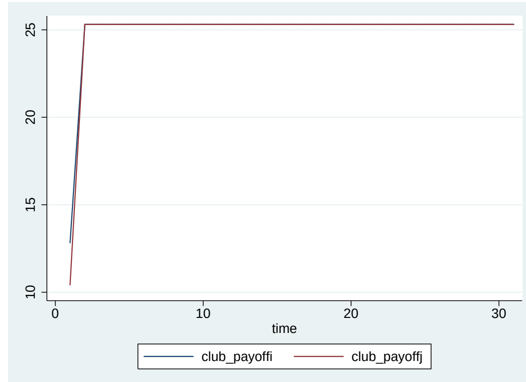

Figure 14: Average club payoff (gift 13), simulation check.

4.3 Parasite recyclers

In this subsection, instead of parasite consumers, the parasite strategy in the simulations used is now parasite recyclers. Parasite recyclers are very similar to parasite consumers in that they both give a gift at the start of a match and defect, but instead of consuming the gift, parasite recyclers perfectly recycle the gift that they were given and re-gift it in the next period to their new partner. Thus, parasite recyclers only have to buy a gift once, in the first period of each parasites existence.

4.3.1 Baseline calibration with parasite recyclers

The first result presented with parasite recyclers uses the same calibration as in Table 2. Because of the nature of parasite recyclers, that they only have to pay for a gift once, one would expect them to have a much better chance at taking over a population. As seen in Figures 15, 16, 17 and 18, the parasites do in fact take over a majority of the population and receive higher payoffs than the average discounted payoff of the agents playing the club strategy. Also, the parasite payoff is higher than the payoff to defecting in every period (4.80 vs. 3.77).

This result lends support to several theories as to why certain types of gift are given. For example, many gifts given at the start of romantic relationships need to be consumed immediately, thus preventing recycling. As well, the custom of removing price tags is very common, which helps to prevent recycling. More on this will be discussed in the next section.

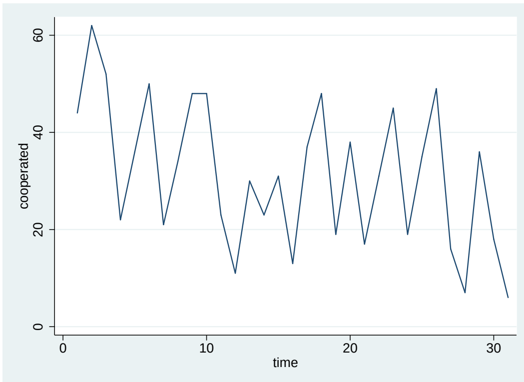

Figure 15: Percentage of agents cooperating with parasite recyclers (gift 13)

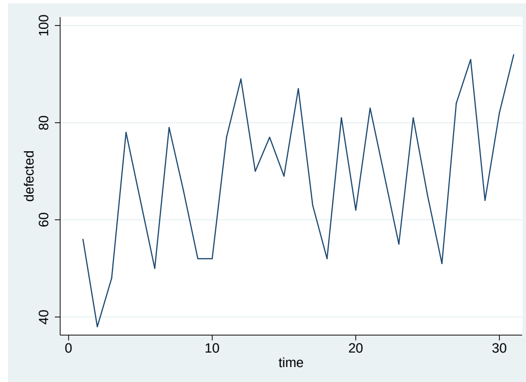

Figure 16: Percentage of agents defecting with parasite recyclers (gift 13)

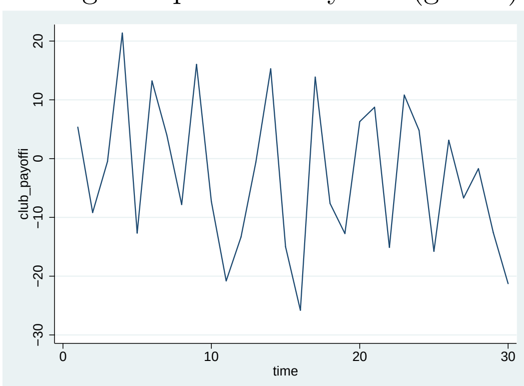

Figure 17: Average club payoff with parasite recyclers (gift 13)

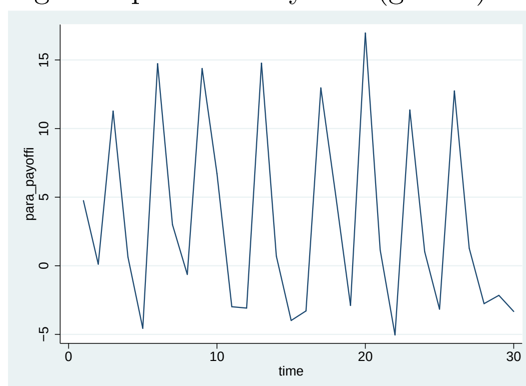

Figure 18: Average parasite payoff with parasite recyclers (gift 13)

4.4 Three strategies

Those to whom we give differ from those to whom we do not give. Those from whom we receive may differ still. Gifts are tangible expressions of social relationships. - Sherry (1983)

In this subsection, there are now two types of clubbies, and parasites are made more powerful. As well, the message sending technology discussed previously is used. In Carmichael and MacLeod, it is hinted that the only strategy that may invade a gift-giving population is a new type of club which coordinates on a lower gift. So, now there are allowed to be low-type clubbies and high-type clubbies. It is assumed that the size of the low-type gift is less than the size of the high-type gift. The low-type clubbies always give a gift. If they are matched with an high-type clubby, they will give the high-type gift. If they are matched with a low-type clubby they will send the low-type gift, and if they are matched with a parasite they will send the low-type gift. If their partner cooperated in the previous round, they will stay matched with that partner until death or the evolutionary step. The high-type clubbies always send the high-type gift and cooperate, and if their partner cooperated they will stay matched with them until death or the evolutionary step. The parasites are now able to perfectly replicate the messages of both types of clubbies, and they are also able to identify other parasites. Thus, if they are matched with a high-type club, they send the high-type gift, defect and quit the match. If they are matched with a low-type club, they send the low-type gift, defect and quit the match. And if they are matched with another parasite they send no gift, defect and quit the match. This is an important change in the parasite behaviour, as if the gifts are too costly, the parasites will be able to do better matching with another parasite in the one-shot game.

4.4.1 Baseline calibration with two types of clubbies

This subsection presents results of the baseline calibration with both low-type and high-type clubbies. The calibration is the same as the previous baseline calibration (Table 4), except now there are low-type and high-type gifts. The high-type clubbies are assumed to be the incumbents, and give a gift of 13. The low-type clubbies give a gift of 10 when matched with another low-type club or with a parasite.

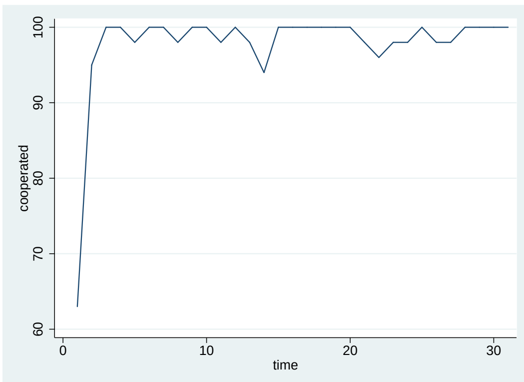

Figure 19: Agents cooperating with two types of clubbies (gifts 13 and 10)

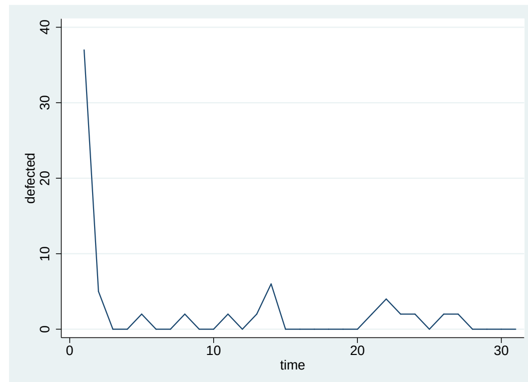

Figure 20: Agents defecting with two types of clubbies (gifts 13 and 10)

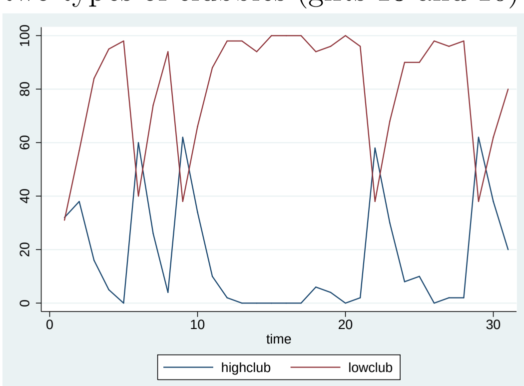

Figure 21: Club strategies with two types of clubbies (gifts 13 and 10)

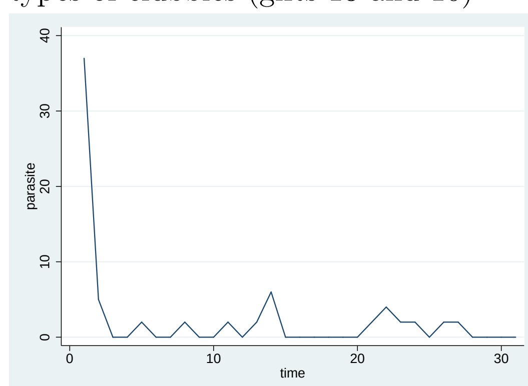

Figure 22: Parasite strategy with two types of clubbies (gifts 13 and 10)

Given the nature of the strategies and the previous simulation with parasite consumers and gifts of 13 (section 4.2.1), one would expect the clubs to take over the population. Because the low-type clubs are allowed to send a smaller gift then the high-type clubs, one would expect the low-type clubs to have a higher average discounted payoff then the high-type clubs and dominate the population.

The results from the simulation are presented in Figures 19 to 22. It can be seen in the figures that the parasite strategy dies out and that the 'low-type' clubbies and the 'high-type' clubbies have a majority of the population, as expected. The reason that the 'high-type' clubbies still remain in the population is because of the strategies: when a low-type club and a high-type club are paired, they exchange the high-type gift. Thus, the only advantage that the low-type club has is when it is paired with another low-type club.

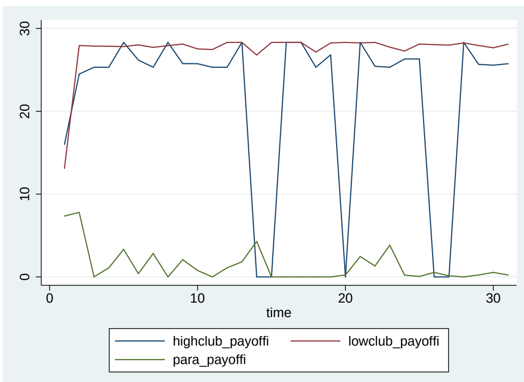

Figure 23: Payoffs for the i-types with two types of clubbies (gifts 13 and 10)

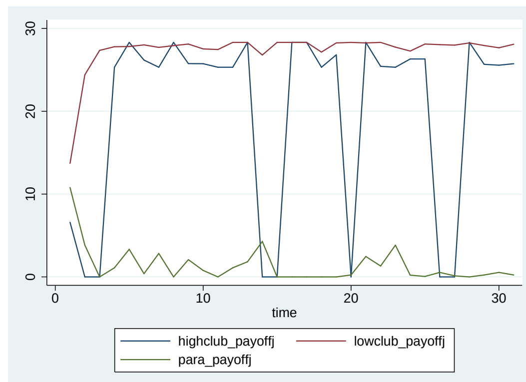

Figure 24: Payoffs for the j-types with two types of clubbies (gifts 13 and 10)

The average discounted payoffs for all the strategies and for both the 'i'-types and 'j'-types can be seen in Figures 23 and 24. Averaging the payoffs from t = 3 to 30, the 'i'-types receive payoffs of: 27.92, 21.75 and 0.97 and the 'j'-types receive payoffs of: 27.90, 20.85 and 0.97 for the low-type clubs, high-type clubs and parasites, respectively. As expected, the average payoffs for the parasites is much lower than for either type of clubs. The low-type clubbies have a higher payoff than the high-type clubbies, as expected. Thus, this simulation confirms what Carmichael and MacLeod suspected, that a strategy which gives a lower gift can take over.

4.4.2 Close to the bound

In this section, the gift sizes are increased to 39 and 36 for the high-type and low-type clubbies, respectively. The purpose of this experiment is to see what would happen if all the strategies have close to the same payoffs. One would expect that if the payoffs from all the strategies were the same, then the population would be randomly determined.

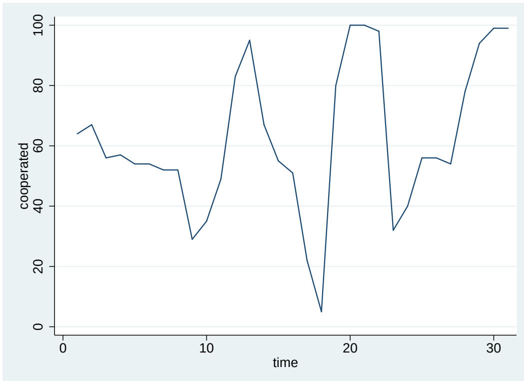

Figure 25: Agents cooperating with two types of clubbies (gifts 39 and 36)

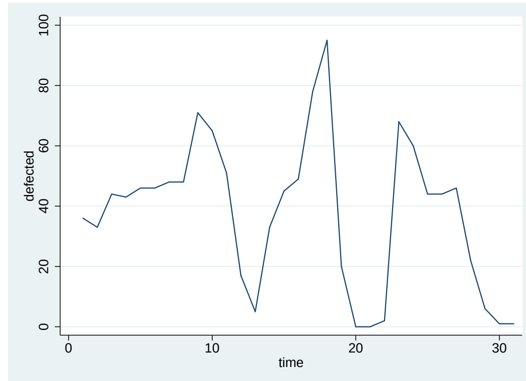

Figure 26: Agents defecting with two types of clubbies (gifts 39 and 36)

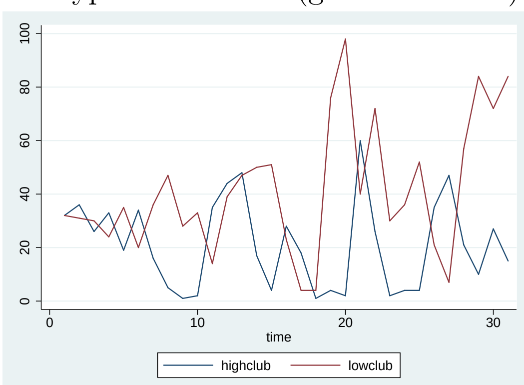

Figure 27: Club strategies with two types of clubbies (gifts 39 and 36)

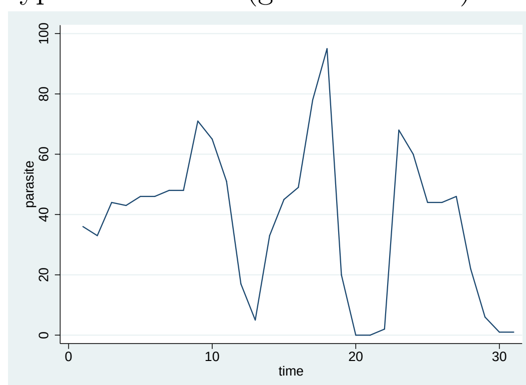

Figure 28: Parasite strategy with two types of clubbies (gifts 39 and 36)

The results from the simulations are presented in Figures 25 to 28. From the figures it can be seen that when all the strategies have close to the same payoffs there appears to be no equilibrium, as expected.

The average discounted payoffs for all the strategies, and both the 'i'-types and 'j'-types are shown in Figures 29 and 30. Averaging from t=3 to 30, the payoffs for the 'i'-types are: -16.50, -27.25 and -12.95, and for the 'j'-types are: -18.29, -19.79 and -12.46 for the low-type clubs, high-type clubs and parasites, respectively. One can see that the average payoffs are below that which would occur if nobody had to give

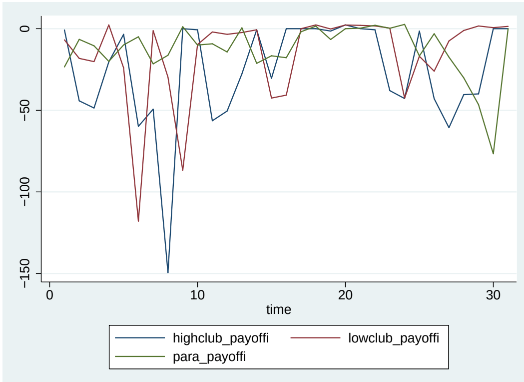

Figure 29: Payoffs for the i-types with two types of clubbies (gifts 39 and 36)

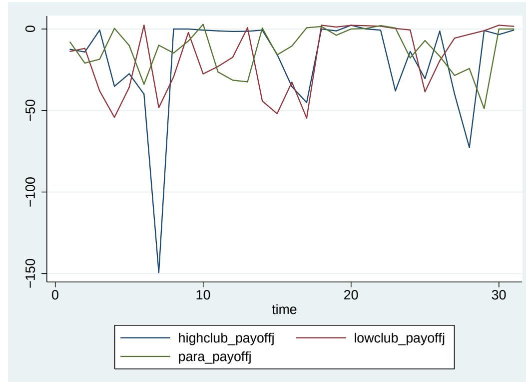

Figure 30: Payoffs for the j-types with two types of clubbies (gifts 39 and 36)

gifts. Thus, if agents had a choice to not give gifts, they probably would choose to opt out of giving. One can see from the previous exercise, if agents can give a smaller gift, then strategies which would be able to give this smaller gift would take over. An interesting extension would be to add in more and more strategies where clubbies could coordinate on smaller and smaller gifts, and then see where the boundary of the smallest gift would be before the payoff to being a parasite matched with a club is so high that parasites take over.

4.4.3 Too big gifts

In this section, the gift sizes are increased even further, to 100 and 97 for the high-type and low-type clubs, respectively. If the gifts given are too high, one would expect the benefits of being a parasite matched with another parasite and not having to give a gift would outweigh the benefits of being a club in a long-term match with another club.

The results of this simulation are shown in Figures 31 to 34. It can be seen that the population eventually converges to almost all parasites. The reason for this is because if a parasite is matched with another parasite, they can now recognize the strategy of their partner through secret messages, and not give a gift and defect. Clubbies always have to give a gift in a new relationship. If the benefits to a club when matched with another club of giving a gift and then remaining in the relationship are smaller than the benefits to parasites, eventually the parasites will take over the population.

In the figures, it appears as if the population is initially at an equilibrium consisting of mostly clubs, and then jumps to another equilibrium consisting of mainly parasites. The reason for this is in the payoffs (see Figures 35 and 36). The parasites initially do very poorly against the population, but then as more clubs enter into the population and match with other clubs, the parasites begin to see the benefits of playing the one-shot game with other parasites and not having to give gifts.

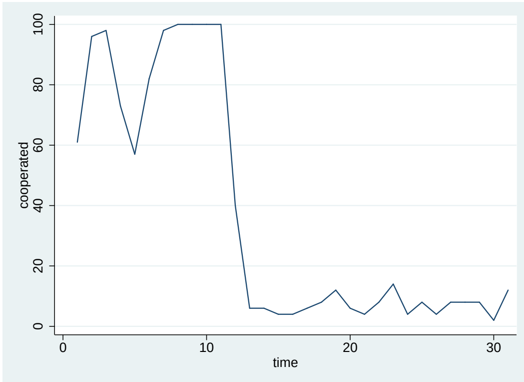

Figure 31: Agents cooperating with two types of clubbies (gifts 100 and 97)

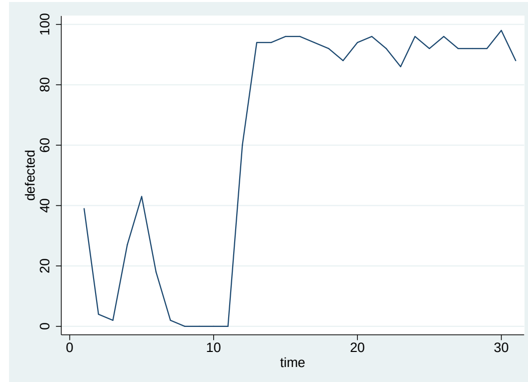

Figure 32: Agents defecting with two types of clubbies (gifts 100 and 97)

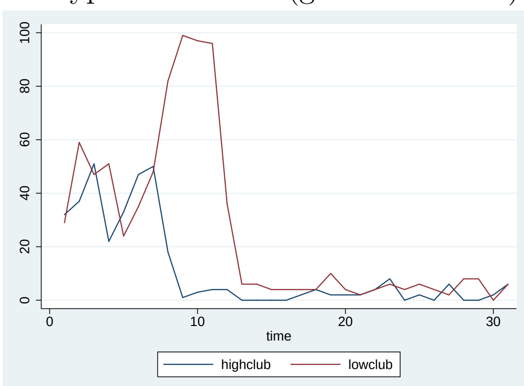

Figure 33: Club strategies with two types of clubbies (gifts 100 and 97)

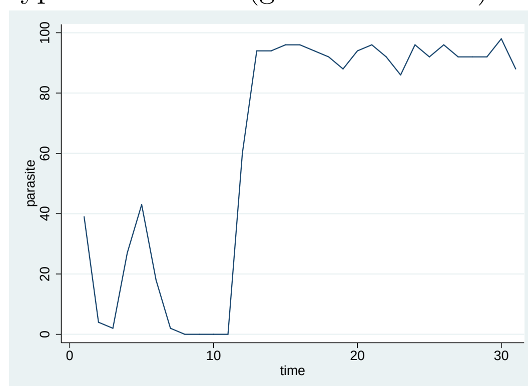

Figure 34: Parasite strategy with two types of clubbies (gifts 100 and 97)

It can be seen from the payoff graphs that the payoffs from being a parasite are higher than for either high-type or low-type clubs. This is because the clubs are entering into long-term relationships, and so the matching market consists mainly of parasites. As discussed previously, the parasites in this case do better when matched with other parasites, which is reflected in their payoff functions. However, as in the previous simulation, all the payoffs are below the payoffs that would occur if nobody gave gifts and defected in every period. Averaging from t = 3 to 30, for the 'i'-types the payoffs are: -56.36, -57.12 and -4.47, and the 'j'-type's payoffs are: -60.04, -50.66 and -14.20, for the low-type clubs, high-type clubs and parasites, respectively.

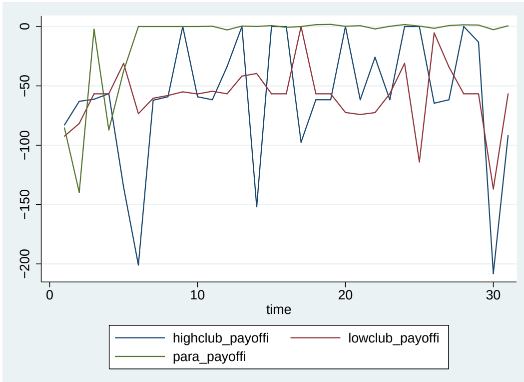

Figure 35: Payoffs for the i-types with two types of clubbies (gifts 100 and 97)

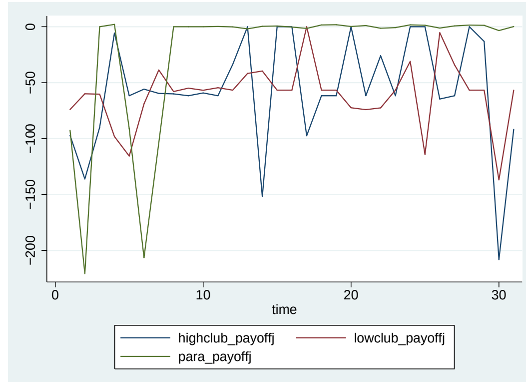

Figure 36: Payoffs for the j-types with two types of clubbies (gifts 100 and 97)

5 Conclusion

In many societies, gifts only surround the long-term relationships. If a relationship is not expected to last very long, cooperation cannot be achieved using gifts. In the model of this paper, this is because all agents would be better off not giving a gift and defecting. As well, the size of the gift and the length of the relationship may be related. In the simulations presented, if the model was iterated more times between evolutionary steps a cooperative outcome may have occurred when the size of the gift was preventing it in the current simulation.

It is necessary that the gifts be inefficient in order for cooperation to emerge. In the symmetric prisoner's dilemma game, if the gifts were not inefficient, the cost to giving the gift would be perfectly offset by the gift received. Examples of inefficient gifts that have emerged in society include chocolates, flowers, and gift baskets of antipasto or muffins. These are all perishable, non-essential items. In the model, the reason perishable goods are often given as gifts can be explained as ways to prevent parasite recyclers. In today's society, gift recycling is still seen occasionally. However, it is a practice that is frowned upon and said to be contrary to the rules of etiquette.

Society has also evolved in other ways to get rid of parasite recyclers. For example, the custom of removing price tags. Because of the removal of price tags, parasite recyclers cannot easily return or exchange a gift. Also, many stores require receipts or the purchaser's credit card to return a gift. This would require making the giver aware of the intention to return the gift, revealing the parasite's strategy. As well, one runs the risk of offending the person who gave it to you. While in the model agents have no history of prior matches, this may not be a realistic assumption in real life.

In this paper, a model was presented and simulated with computational agents. The simulations were run in order to examine an evolutionary process with agents playing a repeated prisoner's dilemma game in which they send messages and/or gifts. When gifts are given of an appropriate size, a stable equilibrium emerges. When robustness tests are presented allowing for parasite recyclers, this outcome breaks down, which may help to explain why specific gifts are given within society. If another strategy is allowed to enter into the economy which coordinates on a smaller gift, it is found that they can grow and thrive in the population. While the simulations may not be entirely realistic, they show that the giving of gifts at the beginning of a relationship can be used to 'glue-together' a match, and the results of the simulations may help to explain why certain types of gifts are given.

Akerlof, G. 1982, *Labor Contracts as Partial Gift Exchange*, The Quarterly Journal of Economics, Vol. 97, pp. 543-569.

Andreoni, J. 1989, *Giving with Impure Altruism: Applications to Charity and Ricardian Equivalence.* The Journal of Political Economy, Vol. 97, No. 6, pp. 1447-1458.

Arrow, K. 1972, *Gifts and Exchanges*, Philosophy and Public Affairs, Vol. 1, No. 4, pp. 343-362.

Axelrod, R. 1984, *The Evolution of Cooperation* New York: Basic Books

Camerer, C. 1988, *Gifts as Economic Signals and Social Signals*, American Journal of Sociology, Vol. 94, pp. S180-S214.

Carmichael, H.L. and MacLeod, W.B. 1997, *Gift Giving and the Evolution of Cooperation.* International Economic Review, Vol. 38, No. 3. pp. 485-509

Harsanyi, J. 1969, *Rational-Choice Models of Political Behavior vs. Functionalist and Conformist Theories* . World Politics, Vol. 21, No. 4, pp. 513-538.

Komter, A. and Vollebergh, W. 1997, *Gift Giving and the Emotional Significance of Family and Friends.* Journal of Marriage and the Family, Vol. 58, No. 3, pp. 747-757.

Kranton, R. 1996, *Reciprocal Exchange: A Self-Sustaining System.* The American Economic Review, Vol. 86, No. 4, pp. 830-851.

Miller, J. 1994, *The coevolution of automata in the repeated prisoner's dilemma.* Journal of Economic Behavior and Organization, Vol 29, pp. 87-112.

Parry, J. (1986), *The Gift, the Indian Gift and the 'Indian Gift'.* Man, New Series, Vol 21. No. 3, pp. 453-473.

Rabin, M. 1993, *Incorporating Fairness into Game Theory and Economics.* The American Economic Review, Vol. 83, pp. 1281-1302.

Ruffle, B. 1999, *Gift giving with emotions.* Journal of Economic Behavior and Organization, Vol. 39, 399-420.

Sherry, J. 1983, *Gift giving in anthropological perspective.* The Journal of Consumer Research, Vol. 10, No. 2, pp. 157-168.

Van Bragt, D., Van Kemenade, C. and La Poutre, H. 2001, *The Influence of Evolutionary Selection Schemes on the Iterated Prisoner's Dilemma.* Computational

Economics, Vol. 17, pp. 253-263.

van de Ven, J. 2000, *The Economics of the Gift.* Department of Economics and CentER, Tillburg University.

A Stata programs

This appendix presents a sample of the programs used to run the simulations. The programs shown are to run simulations with two types of clubbies. Prison3strat.do is the main program. It defines the parameters of the model, and loops through the evolutionary process, calling upon strategies3.do, payoffs33strat.do, death.do, matching3strat.do and evolve3strat.do. For further information on the programs, or for access to the programs used in the other simulations presented, please contact the author.

prison3strat.do

clear all
capture log close
log using “C:\Users\Amy\Desktop\maessay\prison3strat.log”, replace
capture log off
capture log on
set maxvar 32767
infile date using ”C:\Users\Amy\Desktop\maessay\dates.txt”
local rnum = 50
number of i’s and j’s
local mnum = 10
number of rounds in between evolution
local evolvesteps = 30
number of evolutions
local rnum1 = ‘rnum’ + 1
keep in 2/‘rnum1’
tsset date, year
local matchcount = 0
local deathcount = 0
local clubcount = 0
local parasitecount = 0
gen roundnum = 0
local roundin = 0
rename date agent
gen rnum = ‘rnum’
number of agents
gen A = 10
gen B = 19
gen maxstrati = “low club”
gen maxstratj = “low club”
a fraction 1-rho agents die
gen rho = 0.75
local rho = rho in 1
gen deathrho = 0.75

```
local deathrho = deathrho in 1
gen mnum = ‘mnum’
number of matches ie rounds
gen evolvesteps = ‘evolvesteps’
gifts
gen highgift = 39
gen lowgift = 36
local giftval = highgift in 1
gen giftbound = 13
gen copyval = 0.4
percentage of new matches copying their parent’s strategy
gen highval = 0.95
gen lowclubprop = 0.33
gen highclubprop = 0.67
percentage of clubs to start in the matching market
gen i = 0
local A = A in 1
local B = B in 1
if 2*‘A’<=‘B’ continue
if ‘B’<=‘A’ continue
if ‘A’<=1 continue
proposition 7: if this gift size is satisfied, there exists a NSS population which send a
gift bounded by the following:
if highgift[1] = 39 {
if (‘A’-1)/(1-‘rho’)<=highgift[1] continue
if highgift[1] <= (‘B’-‘A’)/(‘rho’) continue
if ‘rho’ <= (‘B’-‘A’)/(‘B’-1) continue
if ‘rho’ >= 1 continue
}
forvalues e = 0/‘evolvesteps’ {
forvalues i = 0/‘mnum’ {
local j = ‘i’ + 1
gen matchi‘j’evolve‘e’ = 0
gen matchj‘j’evolve‘e’ = -1
gen death‘i’evolve‘e’ = ”alive”
gen payoffi‘i’evolve‘e’ = 0
gen payoffj‘i’evolve‘e’ = 0
gen strati‘i’evolve‘e’ = ”nothing”
gen stratj‘i’evolve‘e’ = ”nothing”
gen matchagei‘i’evolve‘e’ = 0
gen matchagej‘i’evolve‘e’ = 0
gen actioni‘i’evolve‘e’ = ”nothing”
gen actionj‘i’evolve‘e’ = ”nothing”
}
}
```

```
local mnum1 = ‘mnum’ + 1
local mnum1 = ‘mnum’ -1
forvalues e = 0/‘evolvesteps’
drop matchi‘mnum1’evolve‘e’
drop matchj‘mnum1’evolve‘e’
}
local evolve = 0
gen evolve = ‘evolve’
while ‘evolve’ ¡= ‘evolvesteps’
gen matchi0evolve‘evolve’ = agent
gen matchj0evolve‘evolve’ = agent
round 0, matched against same agent number
forvalues i = 1/‘rnum’ {
forvalues j = 1/‘rnum’ {
if ‘i’ == ‘j’ {
display ”match”
replace matchi0evolve‘evolve’ = ‘i’ in ‘i’
replace matchj0evolve‘evolve’ = ‘j’ in ‘j’
}
}
}
forvalues i = 0/‘mnum’
replace evolve = ‘evolve’
display ”choose strategy, which is the collection of actions, including separation decisions” and messages
do ”C:\Users\Amy\Desktop\maessay\strategies3strat.do”
do ”C:\Users\Amy\Desktop\maessay\payoffs33strat.do”
display ”check if players are dead and replace dead players”
do ”C:\Users\Amy\Desktop\maessay\death.do”
local roundin = ‘roundin’ + 1
replace roundnum = ‘roundin’
display ‘roundin’
display ‘evolve’
display ”repeat, only matching new and separated players”
if ‘roundin’ > ‘mnum’ continue
else {
do ”C:\Users\Amy\Desktop\maessay\matching3strat.do”
}
} * calculate average discounted payoff and evolve
do ”C:\Users\Amy\Desktop\maessay\evolve3strat.do”
local roundin = 0
replace roundnum = ‘roundin’
local evolve = ‘evolve’ + 1
}
```

strategies3strat.do

local rnum = rnum in 1
local mnum = roundnum in 1
local tmnum = mnum in 1
disp ‘tmnum’
local mnum1 = roundnum-1 in 1
local highgift = highgift in 1
local lowgift = lowgift in 1
local copyval = copyval in 1
local highval = highval in 1
local lowclubprop = lowclubprop in 1
local highclubprop = highclubprop in 1
local evolve = evolve in 1
local evolve1 = evolve-1 in 1
disp ‘evolve1’
if ‘mnum’ == 0 & ‘evolve’ == 0 {
forvalues i = 1/‘rnum’ {
local uniformvali = uniform()
local uniformvalj = uniform()
if ‘uniformvali’ <‘lowclubprop’ {
club strategy
replace strati‘mnum’evolve‘evolve’ = ”low club” in ‘i’
display ”low club strategy for i”
replace i = ‘i’
} else if ‘uniformvali’ >= ‘lowclubprop’ & ‘uniformvali’ < ‘highclubprop’ {
replace strati‘mnum’evolve‘evolve’ = ”high club” in ‘i’
display ”high club strategy for i”
replace i = ‘i’

else
parasite strategy
replace strati‘mnum’evolve‘evolve’ = ”parasite” in ‘i’
display ”parasite strategy for i”
replace i = ‘i’
}
if ‘uniformvalj’ < ‘lowclubprop’ {
club strategy
replace stratj‘mnum’evolve‘evolve’ = ”low club” in ‘i’
display ”club strategy for j”
replace i = ‘i’
}
else if ‘uniformvalj’ > ‘lowclubprop’ & ‘uniformvalj’ < ‘highclubprop’ {
replace stratj‘mnum’evolve‘evolve’ = ”high club” in ‘i’
display ”high club strategy for j”

```
replace i = `i'
}
else {
parasite strategy
replace stratj`mnum'evolve`evolve' = "parasite" in `i'
display "parasite strategy for j"
replace i = `i'
}
}
}
else if `mnum' == 0 & `evolve'  = 0 {
forvalues i = 1/`rnum' {
local uniformval = uniform()
if `uniformval' <= `copyval' {
copy parents strategy
disp "copying parents strategy"
replace strati`mnum'evolve`evolve' = strati`tmnum'evolve`evolve1'[`i'] in `i'
replace stratj`mnum'evolve`evolve' = stratj`tmnum'evolve`evolve1'[`i'] in `i'
}
else if `copyval'<`uniformval' & `uniformval'<=`highval' {
mimic best strategy
disp "mimicking"
replace strati`mnum'evolve`evolve' = maxstrati in `i'
replace stratj`mnum'evolve`evolve' = maxstratj in `i'
}
else {
disp "random"
local uval = uniform()
if `uval'<=0.33 {
replace strati`mnum'evolve`evolve' = "high club" in `i'
replace stratj`mnum'evolve`evolve' = "high club" in `i'
}
else if `uval'>0.33 & `uval' <0.67 {
replace strati`mnum'evolve`evolve' = "low club" in `i'
replace stratj`mnum'evolve`evolve' = "low club" in `i'
else {
replace strati`mnum'evolve`evolve' = "parasite" in `i'
replace stratj`mnum'evolve`evolve' = "parasite" in `i'
}
}
}
}
}
else {
forvalues i = 1/`rnum' {
```

```
if death`mnum'evolve`evolve' == "reincarnated" in `i' {
local uniformval = uniform()
disp `uniformval'
if `uniformval' <= `copyval' {
copy parents strategy
replace strati`mnum'evolve`evolve' = strati`mnum1'evolve`evolve'[`i'] in `i'
replace stratj`mnum'evolve`evolve' = stratj`mnum1'evolve`evolve'[`i'] in `i' }
else if `copyval'<`uniformval' & `uniformval'<=`highval'
disp "mimicking!!!"
replace strati`mnum'evolve`evolve' = maxstrati in `i'
replace stratj`mnum'evolve`evolve' = maxstratj in `i'
}
else {
random strategy
local unival = uniform()
if `unival'<0.34
replace strati`mnum'evolve`evolve' = "low club" in `i'
replace stratj`mnum'evolve`evolve' = "low club" in `i'
display "club strategy for i and j"
replace i = `i'
}
else if `unival' >= 0.34 & `unival' < 0.67
replace strati`mnum'evolve`evolve' = "high club" in `i'
replace stratj`mnum'evolve`evolve' = "high club" in `i'
display "high club strategy for i and j"
replace i = `i'
}
else {
replace strati`mnum'evolve`evolve' = "parasite" in `i'
replace stratj`mnum'evolve`evolve' = "parasite" in `i'
display "parasite strategy for i and j"
replace i = `i'
}
}
}
else {
replace strati`mnum'evolve`evolve' = strati`mnum1'evolve`evolve'[`i'] in `i'
replace stratj`mnum'evolve`evolve' = stratj`mnum1'evolve`evolve'[`i'] in `i'
}
}
}
forvalues i = 1/`rnum' {
if strati`mnum'evolve`evolve' == "nothing" in `i' {
replace strati`mnum'evolve`evolve' = "low club" in `i'
}
```

```
if stratj`mnum'evolve`evolve' == "nothing" in `i' {
replace stratj`mnum'evolve`evolve' = "low club" in `i'
}
}
```

payoffs33strat.do

```
local A = A in 1
local B = B in 1
local rnum = rnum in 1
local mnum = roundnum in 1
local mnum1 = roundnum-1 in 1
local evolve = evolve in 1
local rho = rho in 1
local highgift = highgift in 1
local lowgift = lowgift in 1
if ‘mnum’==0 {
forvalues k = 1/‘rnum’ {
forvalues j = 1/‘rnum’ {
forvalues i = 1/‘rnum’ {
if matchi‘mnum’evolve‘evolve’[‘i’]==‘k’ & matchj‘mnum’evolve‘evolve’[‘j’]==‘k’ {
if strati‘mnum’evolve‘evolve’[‘i’] == ”high club” {
if stratj‘mnum’evolve‘evolve’[‘j’] == ”high club” {
replace payoffi‘mnum’evolve‘evolve’ = ‘A’-‘highgift’ in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
replace payoffj‘mnum’evolve‘evolve’= ‘A’- ‘highgift’ in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
if stratj‘mnum’evolve‘evolve’[‘j’] == ”low club” {
replace payoffi‘mnum’evolve‘evolve’ = ‘A’-‘highgift’ in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
replace payoffj‘mnum’evolve‘evolve’= ‘A’- ‘highgift’ in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
if stratj‘mnum’evolve‘evolve’[‘j’] == ”parasite” {
replace payoffi‘mnum’evolve‘evolve’ = 0 - ‘highgift’ in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
replace payoffj‘mnum’evolve‘evolve’ = ‘B’ - ‘highgift’ in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”defect” in ‘j’
}
}
if strati‘mnum’evolve‘evolve’[‘i’] == ”low club” {
if stratj‘mnum’evolve‘evolve’[‘j’] == ”high club” {
replace payoffi‘mnum’evolve‘evolve’ = ‘A’-‘highgift’ in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
replace payoffj‘mnum’evolve‘evolve’= ‘A’- ‘highgift’ in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
if stratj‘mnum’evolve‘evolve’[‘j’] == ”low club” {
replace payoffi‘mnum’evolve‘evolve’ = ‘A’-‘lowgift’ in ‘i’
```

```
replace actioni`mnum'evolve`evolve' = "cooperate" in `i'
replace payoffj`mnum'evolve`evolve'= `A'- `lowgift' in `j'
replace actionj`mnum'evolve`evolve' = "cooperate" in `j'
}
if stratj`mnum'evolve`evolve'[`j'] == "parasite" {
replace payoffi`mnum'evolve`evolve' = 0 - `lowgift' in `i'
replace actioni`mnum'evolve`evolve' = "cooperate" in `i'
replace payoffj`mnum'evolve`evolve' = `B' - `lowgift' in `j'
replace actionj`mnum'evolve`evolve' = "defect" in `j'
}
}
if strati`mnum'evolve`evolve'[`i'] == "parasite" {
if stratj`mnum'evolve`evolve'[`j'] == "high club" {
replace payoffi`mnum'evolve`evolve' = `B' - `highgift' in `i'
replace actioni`mnum'evolve`evolve' = "defect" in `i'
replace payoffj`mnum'evolve`evolve' = 0 - `highgift' in `j'
replace actionj`mnum'evolve`evolve' = "cooperate" in `j'
}
if stratj`mnum'evolve`evolve'[`j'] == "low club" {
replace payoffi`mnum'evolve`evolve' = `B' - `lowgift' in `i'
replace actioni`mnum'evolve`evolve' = "defect" in `i'
replace payoffj`mnum'evolve`evolve' = 0 - `lowgift' in `j'
replace actionj`mnum'evolve`evolve' = "cooperate" in `j'
}
if stratj`mnum'evolve`evolve'[`j'] == "parasite" {
replace payoffi`mnum'evolve`evolve' = 1 in `i'
replace actioni`mnum'evolve`evolve' = "defect" in `i'
replace payoffj`mnum'evolve`evolve' = 1 in `j'
replace actionj`mnum'evolve`evolve' = "defect" in `j'
}
}
}
}
}
}
}
else {
forvalues k = 1/`rnum' {
forvalues i = 1/`rnum' {
forvalues j = 1/`rnum' {
if matchi`mnum'evolve`evolve'[`i']==`k' & matchj`mnum'evolve`evolve'[`j']==`k' {
if strati`mnum'evolve`evolve'[`i'] == "high club" {
if stratj`mnum'evolve`evolve'[`j'] == "high club" {
if matchagei`mnum'evolve`evolve'[`i'] =0 {
local oldpayoffi = payoffi`mnum1'evolve`evolve' in `i'
```

```
local oldpayoffj = payoffj‘mnum1’evolve‘evolve’ in ‘j’
replace payoffi‘mnum’evolve‘evolve’ = ‘oldpayoffi’ + (‘rho’^‘mnum’)*‘A’ in ‘i’
replace payoffj‘mnum’evolve‘evolve’ = ‘oldpayoffj’ + (‘rho’^‘mnum’)*‘A’ in ‘j’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
else {
local oldpayoffi = payoffi‘mnum1’evolve‘evolve’ in ‘i’
local oldpayoffj = payoffj‘mnum1’evolve‘evolve’ in ‘j’
if death‘mnum’evolve‘evolve’[‘i’] == ”reincarnated” {
replace payoffi‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(‘A’-‘highgift’) in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’] == ”reincarnated” {
replace payoffj‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(‘A’-‘highgift’) in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
if death‘mnum’evolve‘evolve’[‘i’] == ”alive” {
replace payoffi‘mnum’evolve‘evolve’ = ‘oldpayoffi’ + (‘rho’^‘mnum’)*(‘A’-‘highgift’) in
‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’] == ”alive” {
replace payoffj‘mnum’evolve‘evolve’ = ‘oldpayoffj’ + (‘rho’^‘mnum’)*(‘A’-‘highgift’) in
‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
}
}
if stratj‘mnum’evolve‘evolve’[‘j’] == ”low club” {
if matchagei‘mnum’evolve‘evolve’[‘i’] =0 {
local oldpayoffi = payoffi‘mnum1’evolve‘evolve’ in ‘i’
local oldpayoffj = payoffj‘mnum1’evolve‘evolve’ in ‘j’
replace payoffi‘mnum’evolve‘evolve’ = ‘oldpayoffi’ + (‘rho’^‘mnum’)*‘A’ in ‘i’
replace payoffj‘mnum’evolve‘evolve’ = ‘oldpayoffj’ + (‘rho’^‘mnum’)*‘A’ in ‘j’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
else {
local oldpayoffi = payoffi‘mnum1’evolve‘evolve’ in ‘i’
local oldpayoffj = payoffj‘mnum1’evolve‘evolve’ in ‘j’
if death‘mnum’evolve‘evolve’[‘i’] == ”reincarnated” {
replace payoffi‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(‘A’-‘highgift’) in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
}
```

```
if death‘mnum’evolve‘evolve’[‘j’] == ”reincarnated” {
replace payoffj‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(‘A’-‘highgift’) in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
if death‘mnum’evolve‘evolve’[‘i’] == ”alive” {
replace payoffi‘mnum’evolve‘evolve’ = ‘oldpayoffi’ + (‘rho’^‘mnum’)*(‘A’-‘highgift’) in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’] == ”alive” {
replace payoffj‘mnum’evolve‘evolve’ = ‘oldpayoffj’ + (‘rho’^‘mnum’)*(‘A’-‘highgift’) in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
}
}
if stratj‘mnum’evolve‘evolve’[‘j’] == ”parasite” {
local oldpayoffi = payoffi‘mnum1’evolve‘evolve’ in ‘i’
local oldpayoffj = payoffj‘mnum1’evolve‘evolve’ in ‘j’
if death‘mnum’evolve‘evolve’[‘i’]==”reincarnated” {
replace payoffi‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(-‘highgift’+ 0) in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’]==”reincarnated” {
replace payoffj‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(-‘highgift’+ ‘B’) in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”defect” in ‘j’
}
if death‘mnum’evolve‘evolve’[‘i’]==”alive” {
replace payoffi‘mnum’evolve‘evolve’ = ‘oldpayoffi’+(‘rho’^‘mnum’)*(-‘highgift’+ 0) in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’]==”alive” {
replace payoffj‘mnum’evolve‘evolve’ = ‘oldpayoffj’+(‘rho’^‘mnum’)*(-‘highgift’ + ‘B’) in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”defect” in ‘j’
}
}
}
if strati‘mnum’evolve‘evolve’[‘i’] == ”parasite” {
if stratj‘mnum’evolve‘evolve’[‘j’] == ”high club” {
local oldpayoffi = payoffi‘mnum1’evolve‘evolve’ in ‘i’
local oldpayoffj = payoffj‘mnum1’evolve‘evolve’ in ‘j’
if death‘mnum’evolve‘evolve’[‘i’]==”reincarnated” {
replace payoffi‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(-‘highgift’+ ‘B’) in ‘i’
```

```
replace actioni‘mnum’evolve‘evolve’ = ”defect” in ‘i’
}
else {
replace payoffi‘mnum’evolve‘evolve’ = ‘oldpayoffi’+(‘rho’ˆ‘mnum’)*(-‘highgift’+ ‘B’)
in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”defect” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’]==”reincarnated” {
replace payoffj‘mnum’evolve‘evolve’ = (‘rho’ˆ‘mnum’)*(-‘highgift’ + 0) in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
else {
replace payoffj‘mnum’evolve‘evolve’ = ‘oldpayoffj’+(‘rho’ˆ‘mnum’)*(-‘highgift’+ 0) in
‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
}
if stratj‘mnum’evolve‘evolve’[‘j’] == ”low club” {
local oldpayoffi = payoffi‘mnum1’evolve‘evolve’ in ‘i’
local oldpayoffj = payoffj‘mnum1’evolve‘evolve’ in ‘j’
if death‘mnum’evolve‘evolve’[‘i’]==”reincarnated” {
replace payoffi‘mnum’evolve‘evolve’ = (‘rho’ˆ‘mnum’)*(-‘lowgift’+ ‘B’) in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”defect” in ‘i’
}
else {
replace payoffi‘mnum’evolve‘evolve’ = ‘oldpayoffi’+(‘rho’ˆ‘mnum’)*(-‘lowgift’+ ‘B’) in
‘i’
replace actioni‘mnum’evolve‘evolve’ = ”defect” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’]==”reincarnated” {
replace payoffj‘mnum’evolve‘evolve’ = (‘rho’ˆ‘mnum’)*(-‘lowgift’ + 0) in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
else {
replace payoffj‘mnum’evolve‘evolve’ = ‘oldpayoffj’+(‘rho’ˆ‘mnum’)*(-‘lowgift’+ 0) in
‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
}
if stratj‘mnum’evolve‘evolve’[‘j’] == ”parasite” {
local oldpayoffi = payoffi‘mnum1’evolve‘evolve’ in ‘i’
local oldpayoffj = payoffj‘mnum1’evolve‘evolve’ in ‘j’
if death‘mnum’evolve‘evolve’[‘i’]==”reincarnated” {
replace payoffi‘mnum’evolve‘evolve’ = (‘rho’ˆ‘mnum’)*(1) in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”defect” in ‘i’
```

```
}
else {
replace payoffi‘mnum’evolve‘evolve’ = ‘oldpayoffi’+(‘rho’^‘mnum’)*(1) in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”defect” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’]==”reincarnated” {
replace payoffj‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(1) in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”defect” in ‘j’
}
else {
replace payoffj‘mnum’evolve‘evolve’ = ‘oldpayoffj’+(‘rho’^‘mnum’)*(1) in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”defect” in ‘j’
}
}
}
if strati‘mnum’evolve‘evolve’[‘i’] == ”low club” {
if stratj‘mnum’evolve‘evolve’[‘j’] == ”high club” {
if matchagei‘mnum’evolve‘evolve’[‘i’] =0 {
local oldpayoffi = payoffi‘mnum1’evolve‘evolve’ in ‘i’
local oldpayoffj = payoffj‘mnum1’evolve‘evolve’ in ‘j’
replace payoffi‘mnum’evolve‘evolve’ = ‘oldpayoffi’ + (‘rho’^‘mnum’)*‘A’ in ‘i’
replace payoffj‘mnum’evolve‘evolve’ = ‘oldpayoffj’ + (‘rho’^‘mnum’)*‘A’ in ‘j’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
else {
local oldpayoffi = payoffi‘mnum1’evolve‘evolve’ in ‘i’
local oldpayoffj = payoffj‘mnum1’evolve‘evolve’ in ‘j’
if death‘mnum’evolve‘evolve’[‘i’] == ”reincarnated” {
replace payoffi‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(‘A’-‘highgift’) in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’] == ”reincarnated” {
replace payoffj‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(‘A’-‘highgift’) in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
if death‘mnum’evolve‘evolve’[‘i’] == ”alive” {
replace payoffi‘mnum’evolve‘evolve’ = ‘oldpayoffi’ + (‘rho’^‘mnum’)*(‘A’-‘highgift’) in
‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’] == ”alive” {
replace payoffj‘mnum’evolve‘evolve’ = ‘oldpayoffj’ + (‘rho’^‘mnum’)*(‘A’-‘highgift’) in
‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
```

```
}
}
}
if stratj‘mnum’evolve‘evolve’[‘j’] == ”low club” {
if matchagei‘mnum’evolve‘evolve’[‘i’] =0 {
local oldpayoffi = payoffi‘mnum1’evolve‘evolve’ in ‘i’
local oldpayoffj = payoffj‘mnum1’evolve‘evolve’ in ‘j’
replace payoffi‘mnum’evolve‘evolve’ = ‘oldpayoffi’ + (‘rho’^‘mnum’)*‘A’ in ‘i’
replace payoffj‘mnum’evolve‘evolve’ = ‘oldpayoffj’ + (‘rho’^‘mnum’)*‘A’ in ‘j’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
else {
local oldpayoffi = payoffi‘mnum1’evolve‘evolve’ in ‘i’
local oldpayoffj = payoffj‘mnum1’evolve‘evolve’ in ‘j’
if death‘mnum’evolve‘evolve’[‘i’] == ”reincarnated” {
replace payoffi‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(‘A’-‘lowgift’) in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’] == ”reincarnated” {
replace payoffj‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(‘A’-‘lowgift’) in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
if death‘mnum’evolve‘evolve’[‘i’] == ”alive” {
replace payoffi‘mnum’evolve‘evolve’ = ‘oldpayoffi’ + (‘rho’^‘mnum’)*(‘A’-‘lowgift’) in
‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’] == ”alive” {
replace payoffj‘mnum’evolve‘evolve’ = ‘oldpayoffj’ + (‘rho’^‘mnum’)*(‘A’-‘lowgift’) in
‘j’
replace actionj‘mnum’evolve‘evolve’ = ”cooperate” in ‘j’
}
}
}
if stratj‘mnum’evolve‘evolve’[‘j’] == ”parasite” {
local oldpayoffi = payoffi‘mnum1’evolve‘evolve’ in ‘i’
local oldpayoffj = payoffj‘mnum1’evolve‘evolve’ in ‘j’
if death‘mnum’evolve‘evolve’[‘i’]==”reincarnated” {
replace payoffi‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(-‘lowgift’+ 0) in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’]==”reincarnated” {
replace payoffj‘mnum’evolve‘evolve’ = (‘rho’^‘mnum’)*(-‘lowgift’+ ‘B’) in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”defect” in ‘j’
```

```
}
if death‘mnum’evolve‘evolve’[‘i’]==”alive” {
replace payoffi‘mnum’evolve‘evolve’ = ‘oldpayoffi’+(‘rho’^‘mnum’)*(-‘lowgift’+ 0) in ‘i’
replace actioni‘mnum’evolve‘evolve’ = ”cooperate” in ‘i’
}
if death‘mnum’evolve‘evolve’[‘j’]==”alive” {
replace payoffj‘mnum’evolve‘evolve’ = ‘oldpayoffj’+(‘rho’^‘mnum’)*(-‘lowgift’ + ‘B’) in ‘j’
replace actionj‘mnum’evolve‘evolve’ = ”defect” in ‘j’
}
}
}
}
}
}
}
}
```

death.do

```
local rnum = rnum in 1
local deathrho = deathrho in 1
local mnum = roundnum in 1
local mnum1 = roundnum + 1 in 1
local tmnum = mnum in 1
local evolve = evolve in 1
forvalues i = 1/‘rnum’ {
forvalues j = 1/‘rnum’ {
if ‘i’==‘j’ {
if ‘mnum1’ > ‘tmnum’ continue
else if uniform() < ((1-‘deathrho’)/2) {
display ”match died”
replace death‘mnum’evolve‘evolve’ = ”died” in ‘i’
replace death‘mnum’evolve‘evolve’ = ”died” in ‘j’
replace death‘mnum1’evolve‘evolve’ = ”reincarnated” in ‘i’
replace death‘mnum1’evolve‘evolve’ = ”reincarnated” in ‘j’
replace matchagei‘mnum1’evolve‘evolve’ = 0 in ‘i’
replace matchagej‘mnum1’evolve‘evolve’ = 0 in ‘j’
replace payoffi‘mnum1’evolve‘evolve’ = 0 in ‘i’
replace payoffj‘mnum1’evolve‘evolve’ = 0 in ‘j’
}
}
}
}
```

matching3strat.do

```
* matching market
gen uniformj = 0
gen uniformi = 0
local roundcount = roundnum in 1
local oldnum = ‘roundcount’ -1
local matchcount = 1
local rnum = rnum in 1
local evolve = evolve in 1
forvalues j = 1/‘rnum’ {
replace uniformj = int(20*uniform()) in ‘j’
}
forvalues i = 1/‘rnum’ {
replace uniformi = int(20*uniform()) in ‘i’
}
forvalues i = 1/‘rnum’ {
forvalues k = 1/‘rnum’ {
if matchi‘oldnum’evolve‘evolve’[‘i’]==matchj‘oldnum’evolve‘evolve’[‘k’] {
if actioni‘oldnum’evolve‘evolve’[‘i’]==”cooperate”
& actionj‘oldnum’evolve‘evolve’[‘k’]==”cooperate” {
replace matchi‘roundcount’evolve‘evolve’ = ‘matchcount’ in ‘i’
replace matchj‘roundcount’evolve‘evolve’ = ‘matchcount’ in ‘k’
replace matchagei‘roundcount’evolve‘evolve’ = matchagei‘oldnum’evolve‘evolve’[‘i’] + 1 in ‘i’
replace matchagej‘roundcount’evolve‘evolve’ = matchagej‘oldnum’evolve‘evolve’[‘k’] + 1 in ‘k’
local matchcount = ‘matchcount’ + 1
}
}
}
}
while ‘matchcount’ <= ‘rnum’ {
forvalues k = 1/‘rnum’ {
forvalues i = 1/‘rnum’ {
if uniformi[‘i’]  = uniformj[‘k’] continue
if matchi‘roundcount’evolve‘evolve’[‘i’]  = 0 continue
if matchj‘roundcount’evolve‘evolve’[‘k’]  = -1 continue
else {
replace matchi‘roundcount’evolve‘evolve’ = ‘matchcount’ in ‘i’
replace matchj‘roundcount’evolve‘evolve’ = ‘matchcount’ in ‘k’
local matchcount = ‘matchcount’ + 1
}
}
}
```

```
forvalues j = 1/`rnum' {
replace uniformj = int(3*uniform()) in `j'
}
forvalues i = 1/`rnum' {
replace uniformi = int(3*uniform()) in `i'
}
}
```

evolve3strat.do

local rnum = rnum in 1
number of agents
local mnum = mnum in 1
round numbers
local evolve = evolve in 1
local highclubi = 0
local highclubj = 0
local lowclubi = 0
local lowclubj = 0
local parai = 0
local paraj = 0
local avehighclubpayoffievolve‘evolve’ = 0
local avehighclubpayoffjevolve‘evolve’ = 0
local avelowclubpayoffievolve‘evolve’ = 0
local avelowclubpayoffjevolve‘evolve’ = 0
local aveparapayoffievolve‘evolve’ = 0
local aveparapayoffjevolve‘evolve’ = 0
forvalues i = 1/‘rnum’ {
if strati‘mnum’evolve‘evolve’[‘i’] == ”high club” {
local avehighclubpayoffievolve‘evolve’ = ‘avehighclubpayoffievolve‘evolve”
+ payoffi‘mnum’evolve‘evolve’[‘i’]
local highclubi = ‘highclubi’ + 1
disp ‘avehighclubpayoffievolve‘evolve”
}
if strati‘mnum’evolve‘evolve’[‘i’] == ”low club” {
local avelowclubpayoffievolve‘evolve’ = ‘avelowclubpayoffievolve‘evolve”
+ payoffi‘mnum’evolve‘evolve’[‘i’]
local lowclubi = ‘lowclubi’ + 1
disp ‘avelowclubpayoffievolve‘evolve”
}
else if strati‘mnum’evolve‘evolve’[‘i’] == ”parasite” {
local aveparapayoffievolve‘evolve’ = ‘aveparapayoffievolve‘evolve”
+ payoffi‘mnum’evolve‘evolve’[‘i’]
local parai = ‘parai’ + 1
}
if stratj‘mnum’evolve‘evolve’[‘i’] == ”high club” {
local avehighclubpayoffjevolve‘evolve’ = ‘avehighclubpayoffjevolve‘evolve”
+ payoffj‘mnum’evolve‘evolve’[‘i’]
local highclubj = ‘highclubj’ + 1
disp ‘avehighclubpayoffjevolve‘evolve”
}
if stratj‘mnum’evolve‘evolve’[‘i’] == ”low club” {
local avelowclubpayoffjevolve‘evolve’ = ‘avelowclubpayoffjevolve‘evolve”

```
+ payoffj‘mnum’evolve‘evolve’[‘i’]
local lowclubj = ‘lowclubj’ + 1
disp ‘avelowclubpayoffjevolve‘evolve”
}
else if stratj‘mnum’evolve‘evolve’[‘i’] == ”parasite” {
local aveparapayoffjevolve‘evolve’ = ‘aveparapayoffjevolve‘evolve”
+ payoffj‘mnum’evolve‘evolve’[‘i’]
local paraj = ‘paraj’ + 1
}
}
if ‘highclubi’ = 0 {
gen avehighclubpayoffievolve‘evolve’ = ‘avehighclubpayoffievolve‘evolve”/‘highclubi’
}
else {
gen avehighclubpayoffievolve‘evolve’ = -1000000
}
if ‘lowclubi’ = 0 {
gen avelowclubpayoffievolve‘evolve’ = ‘avelowclubpayoffievolve‘evolve”/‘lowclubi’
}
else {
gen avelowclubpayoffievolve‘evolve’ = -1000000
}
if ‘parai’ = 0 {
gen aveparapayoffievolve‘evolve’ = ‘aveparapayoffievolve‘evolve”/‘parai’
}
else {
gen aveparapayoffievolve‘evolve’ = -1000000
}
if ‘highclubj’ = 0 {
gen avehighclubpayoffjevolve‘evolve’ = ‘avehighclubpayoffjevolve‘evolve”/‘highclubj’
}
else {
gen avehighclubpayoffjevolve‘evolve’ = -1000000
}
if ‘lowclubj’ = 0 {
gen avelowclubpayoffjevolve‘evolve’ = ‘avelowclubpayoffjevolve‘evolve”/‘lowclubj’
}
else {
gen avelowclubpayoffjevolve‘evolve’ = -1000000
}
if ‘paraj’ = 0 {
gen aveparapayoffjevolve‘evolve’ = ‘aveparapayoffjevolve‘evolve”/‘paraj’
}
else {
gen aveparapayoffjevolve‘evolve’ = -1000000
```

```
}
if aveparapayoffievolve`evolve'[1]>avehighclubpayoffievolve`evolve'[1]
& aveparapayoffievolve`evolve'[1]>avelowclubpayoffievolve`evolve'[1] {
replace maxstrati = "parasite"
}
else if avehighclubpayoffievolve`evolve'[1]>avelowclubpayoffievolve`evolve'[1]
& avehighclubpayoffievolve`evolve'[1]>aveparapayoffievolve`evolve'[1] {
replace maxstrati = "high club"
}
else {
replace maxstrati = "low club"
}
if avehighclubpayoffjevolve`evolve'[1]>aveparapayoffjevolve`evolve'[1]
& avehighclubpayoffjevolve`evolve'[1]>avelowclubpayoffjevolve`evolve'[1] {
replace maxstratj = "high club"
}
else if aveparapayoffjevolve`evolve'[1]>avehighclubpayoffjevolve`evolve'[1]
& aveparapayoffjevolve`evolve'[1]>avelowclubpayoffjevolve`evolve'[1] {
replace maxstratj = "parasite"
}
else {
replace maxstratj = "low club"
}
```