Housemate game

Uzi Friedman

**Abstract**

In this paper I examine the different possible strategies when playing a type of extended chicken game, which involves both sides choosing when, not if, they are going to yield. I predict that it will be difficult to find a strategy which will consistently be more effective than yielding at the first chance possible. This assumption relays on similar papers on the difficulty of finding a “best strategy” for iterated Prisoner’s Dilemma and Snowdrift games[[1]](#footnote-1). Experimental results show that under some conditions, such a strategy is found, but its effectiveness is not much greater than that of the naïve strategy.

**Introduction**

The housemate game is a two player simultaneous game based on similar games such as the game of chicken or the snowdrift game. Both players represent two people who share a house and have to individually decide when to complete communal tasks, such as taking our the trash or washing the dishes. Both players want to see the dishes cleaned but neither want to go to the effort of washing them. The dishes being cleaned benefits both players, but the more time goes on without them being taken care of both suffer (smells, lack of sink space, etc). In the traditional game of chicken, both players decide simultaneously between continuing and yielding (see payoff table in figure 1). The principle of the game is that while it is to both players’ benefit if one player yields, the other player's optimal choice depends on what his opponent is doing: if his opponent yields, the player should not, but if the opponent fails to yield, the player should. There are a lot of papers on possible strategies for iterated versions of these kind of games but in this paper we try to take a look at one version in which players don’t choose whether to defect, but when to do so.

In our version of the game, instead of choosing between two options, each player chooses an integer which represents how long he\she is willing to wait before completing a given communal task – when to yield. The payoff for the completion of this task decays as time passes. The amount by which the value decreases is given as a decay constant and is calculated as such:

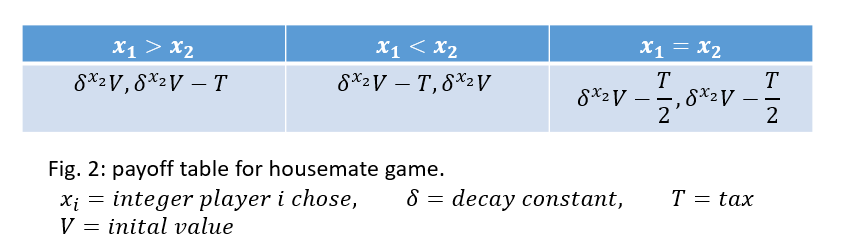
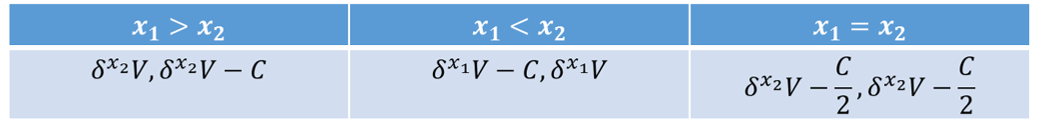
Where the “Time” variable is the smaller of the two integers chosen by the players. Choosing the smaller integers means yielding first, and therefor having to pay a “Tax”, a deduction of a fixed amount of points. This deduction represents the effort involved in completing the task (see payoff figure 2).

The main purpose of this paper is to explore how the introduction of time into the game influenced it, find how human players approach it, and find a viable strategy which is consistently better than the naïve one.

In order to preform experiments, I built a web application that allows players to play the iterated version of the new format. The application allowed me to find the general patterns of play found among human players, test AI agents that implemented those strategies and gather data on their effectiveness. Results showed that most human players’ strategies could be generalized into a few broad categories –

* Conservative players who prefer to stay on the safe side and pick very small integers, even when the decay factor is small in comparison with the tax.
* Greedy players who preferred to choose integers very close to the highest possible one (without risking negative payoff), even when the decay factor is large in comparison with the tax.
* Players who attempted the “tit-for-tat” strategy.
* Players who attempted to “one-up” their competitor in each following round.

|  |  |  |
| --- | --- | --- |
|  | Swerve | Straight |
| Swerve | 0,0 | 0,1 |
| Straight | 1,0 | -10,-10 |
| Fig 1: chicken payoff | | |



**Application & game rules**

To explore this version of the game I built a web application using oTree, a software platform for economics experiments[[2]](#footnote-2). That allowed players to play the game with the new rules and allows automated agents to compete against one another. Players would receive a link which would match them with an opponent and shown a screen with instructions and a blank box to enter an integer. Players would play each opponent for 5 rounds, with the current round number and total round numbers appearing promptly on the top of the page (see figure 3). After entering their choice, they would go into a waiting screen until their opponent made his choice and then returned to the previous screen with an updated round number and a table indicating what were both players’ choices and their payoff for the round (figure 4). After all five rounds were concluded each player got a new link assigning him to a new opponent.

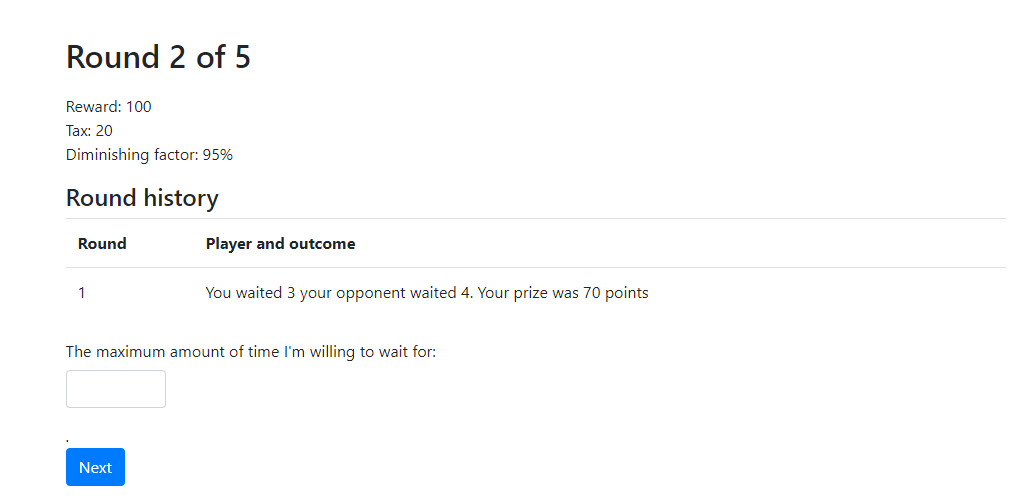
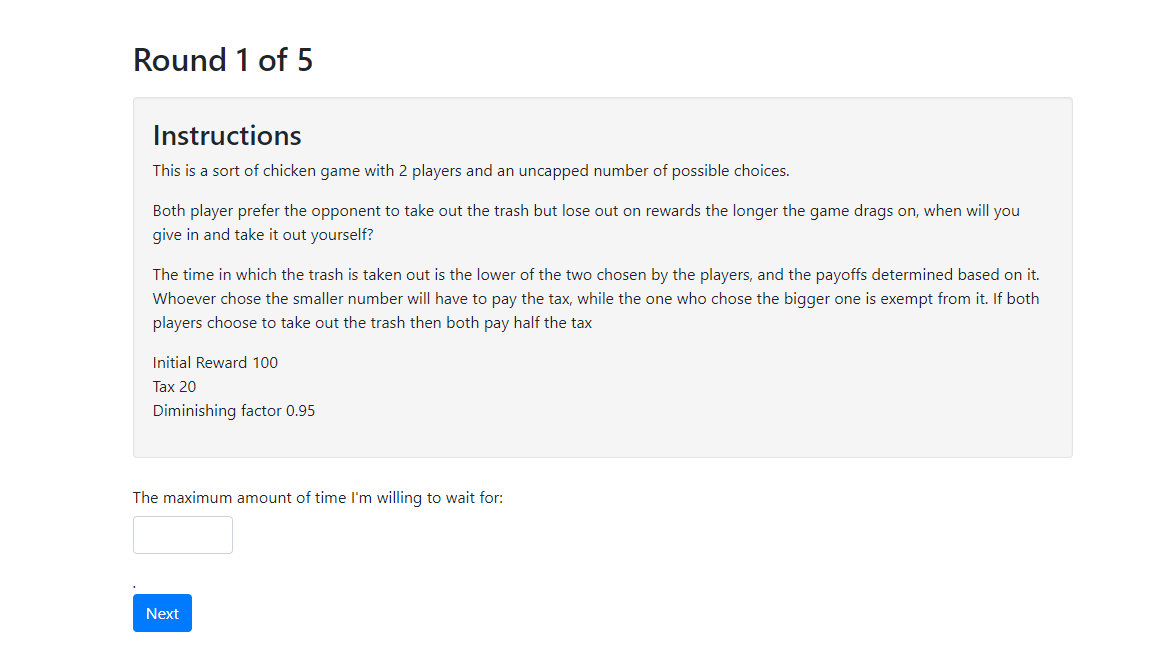


Fig. 4: application’s screen during match

Fig. 3: opening screen of application

**Experimental design and procedures**

The experiment was conducted at the computer lab of BGU. The study group consisted of five men and one woman, all around the ages of 25-35. All participants were given a computer and assigned an unknown opponent. During the experiment participants were asked to refrain from talking to each other, as to avoid comparing strategies. After the first five minutes of starting the game new links were sent to all participants so they could be matched to a new opponent. This procedure was repeated four times, so each player played a total of 20 rounds[[3]](#footnote-3). After all the results were analyzed and a basic understanding of player types was formed, these types were implemented as basic AI agents. In addition to those based on observed strategies two additional agents were created

* A semi random agent who each round ran a loop which started chose an integer at random, with a higher chance of choosing integers within a certain range.
* An agent who was an improvement of the “one-upper”. It would decide if to one-up the opponents last picked integer or to choose 0, whichever would produce the higher payoff assuming the opponent would pick the same integer this round.

Each agent played against all six different agents, five rounds each. The results of this tournament were gathered and the process was repeated with different parameter values.

**Results**

Unsurprisingly, the conservative strategy was able to consistently perform well and was not affected by changes in the decay constant (figure 5). On the other hand, while the greedy agent was the weakest preforming one, with an average payoff of around 75% of that of the conservative agent, it seemed to improve as the decay constant was increased.

The improved one-upper agent was able to outperform the conservative one in most scenarios and got increasingly relatively better as the tax went up (figure 6) and unchanged by increase in the decay constant (figure 5).

Fig. 5: comparison of agents’ performance with changes in decay factor

Fig. 6: comparison of agents’ performance with changes in tax

**discussion**

Clearly there is still a great deal of work that can be done on this subject. While this paper attempted to display some of the properties of the iterated game and the effects of adding time as a variable there are many points lefts untouched. The improved one-upper strategy discussed here is by no means the best possible strategy, and at no point were different compositions of players taken into consideration. All agents presented here only took into consideration their opponents last action and their performance could possibly be greatly improved by expanding their memories to earlier rounds as well. Further work could be done to research how these agents interact with human players and compare those findings to the rankings found here.

1. Christian Hilbe, Arne Traulsen, Karl Sigmund. “Partners or rivals? Strategies for the iterated prisoner's dilemma”

   William H. Press and Freeman J. Dyson. “Iterated Prisoner’s Dilemma contains strategies that dominate any evolutionary opponent”

   Rolf Kümmerli, Caroline Colliard, Nicolas Fiechter, Blaise Petitpierre, Flavien Russier and Laurent Keller. “Human cooperation in social dilemmas: comparing the Snowdrift game with the Prisoner's Dilemma” [↑](#footnote-ref-1)
2. http://otree.readthedocs.io/en/latest/ [↑](#footnote-ref-2)
3. During all human games the constants were [↑](#footnote-ref-3)