Doctoral Thesis

University of Trento

School of Social Sciences

Doctoral School in Economics and Management
Sociology and Social Research
Local Development and Global Dynamics

Society in simulation
On conflict & negotiation of conflict

a dissertation submitted to the doctoral school of economics and management / sociology and social research / local development and global dynamics in partial fulfillment of the requirements for the Doctoral degree (Ph.D.) in Economics and Management

Linh Chi Nguyen

August 2, 2019

Advisor:
Professor Edoardo Gaffeo

Doctoral Committee:
TBD

Society in simulation
On repeated Nash Demand game & bargaining behavior

Contents

1 Literature review **4**

2 The game and simulation configuration **10**
2.1 The game . . . 10
2.2 The problem . . . 11
2.3 Evolutionary algorithm . . . 11
2.4 Payoff calculation . . . 12
2.5 500 rounds . . . 13
2.6 Strategy implementation: finite state machine . . . 13
2.6.1 More examples . . . 14
2.7 How to interpret the result . . . 15
2.7.1 Evolutionarily stable strategy . . . 16

3 Simulation result: the repeated Nash Demand Game **18**
3.1 Patient players . . . 18
3.1.1 $\delta = 0.9$ . . . 18
3.2 Characteristic evaluation and interpretation . . . 26
3.2.1 $\delta = 0.9$ . . . 31
3.2.2 $\delta = 0.8$ . . . 33
3.3 Myopic players . . . 35
3.3.1 $\delta = 0.5$ . . . 35
3.4 Conclusion . . . 36

4 The repeated Nash Demand game with a different approach: Markov chain in strategy structure **38**
4.1 Strategy implementation . . . 38
4.1.1 Classic automata . . . 39
4.2 Strategy interpretation . . . 40
4.2.1 Benchmark . . . 40
4.2.2 . . . 41

5 Simulation result: the repeated Nash Demand Game - Markov chain **42**
5.1 Patient players . . . 42
5.1.1 $\delta = 0.99$ . . . 42
5.1.2 $\delta = 0.95$ . . . 52
5.1.3 $\delta = 0.9$ . . . 53
5.2 Individuals with moderate patience . . . 54
5.2.1 $\delta = 0.8$ . . . 54
5.2.2 $\delta = 0.7$ . . . 56
5.3 Myopic players . . . 56
5.3.1 $\delta = 0.5$ . . . 56
5.4 Conclusion . . . 57

Acknowledgement

I thank professor Erdoardo Gaffeo, two referees and the School of Social Science, together with many people that got out of their way to provide support for me and my research.

Abstract

I simulate a population of agents playing the repeated Nash Demand game. The agents adopt strategies from the infinite strategy space and the population evolves over cycles in a survival of the fittest fashion. Certain strategies become stable during the simulation, causing different levels of average payoff.[1]

[1] I assume the validity of this body of research in the literature.

Chapter 1

Literature review

The literature on game theory and simulation work is vast. I try to present several important work that I deem relevant to show how this body of research has evolved and how my work might fit in with the big picture. The reference list even includes work that are not cited, however, play an important part in shaping the narration of this thesis. The interested audience, therefore, can browse them for a better understanding of the field. Despite the effort to create a coherent vision, with limited resource and capability, I apologise for any mistake the audience might find in this thesis. To sum up briefly, I would visit the work of Maynard Smith and Price (1970s) on evolutionary game theory, then Axelrod (1980s), Rubinstein and Binmore (late 1980s) and Nowak and co. (1990s) on population game simulation. Then I consider two papers on the repeated Nash demand game by Peyton Young and co. (2000s), together with some work by J. Halpern (2010s) on agents and resource sharing. The order of the papers is mostly chronological and a sub-criteria is that I tie similarly inspired papers into clusters. I believe that listing them as data points would make it easier to envision a thread of discussion that spans decades, about logic, conflict negotiation and humanity.

I start with the paper on The logic of animal conflict by Maynard Smith and Price in 1973. Their work traces back to more work before them. It shows that Nature [1] hints biologists about Nash equilibrium since the work of Fisher about the extraordinary sex ratio of 1:1. Hamilton (1967) calls the ratio 1:1 unbeatable and makes cases for several other equilibrium scenarios with unexploitable sex ratio. Maynard Smith and Price (1973) base on the mathematical work of Mac Arthur and the biological investigation of Hamilton, generalize phenotypes into coded strategies. They start to let distinct interests use these strategies to play the game. They also go further to define the term Evolutionarily stable strategy. This notion can be considered a refinement of the Nash equilibrium.

Here is the set up of their simulation: They consider the situation of an animal having conflicts with others in the animal society. The most obvious conflicts concern food and mate and territory. Which can be considered as they fight for their reproductive fitness. To put it simply, the animal have two tactics to choose from: a conventional tactic C which limits the use of physical violence, for example: to display threat at a distance, and a dangerous tactic D which embraces fully the intention to kill. Tactic D has a fixed probability of serious injuring opponent. Whoever is badly injured always retreats, ending the war for good. Furthermore, whoever plays D first is described as using a probe or a provocation. A probe after the opening move of a C is called an escalation. And to play D in return for a D is to retaliate.

Given these reasoning, the animal would consider mainly 3 factors. These factors are factored in the payoff calculation as follows:

- The advantage of winning: +60 points.
- The disadvantage of being seriously injured: - 100 points.
- The disadvantage of wasting time and energy: awarded points ranging from 0 to 20.

They consider 5 major strategies to be representative:

[1] reportedly

	Mouse	Hawk	Bully	Retaliator	Prober-retaliator
Mouse	29	19.5	19.5	29	17.2
Hawk	80	-19.5	74.6	-18.1	-18.9
Bully	80	4.9	41.5	11.9	11.2
Retaliator	29	-22.3	57.1	29	23.1
Prober-retaliator	56.7	-20.1	59.4	26.9	21.9

Table 1.1: Maynard Smith and Price's representative strategies.

- Mouse: always play C. This strategy exists to represent a certain fraction of the population that is young, old and vulnerable.
- Hawk: always play D. This is a total war strategy. The rest is limited war strategies.
- Bully: play D in the first move or in response to C. Play C in response to D. Retreat after being attacked twice
- Retaliator: play C but return a D with a D with high probability
- Prober-retaliator: play C with a low probability of probing. Play C after being retaliated but keep playing D if not being retaliated. Always retaliate.

Stopping rules:
- After a certain number of rounds
- After one contestant retreats

The contests then would be run in pair among the 5 types for many times in order to calculate the stable payoff of each strategy. The result is displayed in a matrix table.

From the table, we can see that Hawk cannot be an ESS because in a population full of Hawk, Mouse and Bully always do better. Regarding a population full of Hawk, it would be easy for Mouse or Bully to invade the population. Mouse is also not an ESS. However, the retaliators can coexist and be ESS.

Using this simulation result, Maynard Smith and Price suggest the interpretation that a total war strategy against ones own kind is not an evolutionarily stable strategy. It would easily lead to extinction of one kind due to individual interest. Hence, to resolve intra conflict, it is natural that one adopts a limited war strategy, for example, displaying of strength instead of using fatal weapons.

The game in this context, though not clearly defined, resembles a repeated prisoners dilemma. And the number of strategies in use, is by no means exhaustive. They, however, are able to represent interesting dynamics of existing behavior patterns in the wild.

The second work I would like to visit, is about population game simulation. Axelrod, in 1980, published his work on Effective choice in the prisoners dilemma. He announced a tournament and invited everyone to send in a program to play the repeated prisoners dilemma. The program inputs the history of previous rounds and outputs the move for the current round. There are, in total, 15 strategies.

Here is the set up:
- Each strategy is paired with each other strategy.
- The pair plays the game for 200 rounds.
- Because a strategy can be probabilistic, each match is run 5 times to estimate a stable payoff for each strategy.
- The benchmark for absolute cooperation is 600 points, for absolute defection is 200 points, though the possible range can be from 0 to 1000.

He then uses the simulation result to describe and comment on the performance of each strategy. The strategy TIT FOR TAT wins the tournament.

He identifies some quality for winning such as:
- Niceness: never the first to defect
- Forgiving: ready to forgive an uncalled for defection
- Retaliate: no hesitate to punish

The approach he uses is very similar to Maynard Smith and Price, in that he collects knowledge-based strategies from experts and hunch-based strategies in the case of laymen. The author then tests these strategies against each other, one at a time. By doing this, he is able to comment on details of each match, the surprise, the sophistication, elegance and effectiveness.

In that same year, he published a second paper on More effective choice in the prisoners dilemma. He did the tournament a second time with the result of the first time published. This time, there were 62 strategies sent in. This time he chose 5 representative strategies and argued that these 5 representatives form a set of strategies that predicts pretty well the performance of a contestant, hence the name representatives.

From this selection pool, TIT FOR TAT comes out to be the winner again. He runs a population with all these strategies but certainly without mutation. TIT FOR TAT wins just as the payoff matrix calculation shows.

In 1987, he published another paper on The evolution of strategies in the iterated prisoners dilemma. This time the work is done more thoroughly. He explored the game in a truly evolutionary setting. He used a Markov process of memory 3 to represent strategy. In the first simulation setting, he makes use of 8 representatives:

Simulation 1:
- Initial population of 20 random individuals.
- Run each random individual with each of 8 representatives.
- Produce the next generation by ranking the scores and gives individuals the accordingly number of matings (for example: 1 for an average score, 2 for a better score..)
- For each of 10 matings, create 2 offsprings from the two selected parents using crossover and mutation
- Run the simulation for 50 generations
- Crossover (i.e. sexual production) is used because it facilitates evolution (Maynard Smith).
- 40 runs of the identical condition is run to determine the variability of results.

From this result, he finds out that the winning rules are pretty similar to TIT FOR TAT. However, they are better. They are able to discriminate among the representatives then exploit the exploitable one, but apologise and cooperate with the unexploitable. These winning rules are different from TIT FOR TAT in the niceness quality: they always defect in the first round, or even in the second round. They then use the response of opponent to evaluate what type of representatives opponent is.

It is to be noted that this is a contest against a fixed set of representatives. The evolutionary force would naturally shapes a creature that fits well with this fixed environment of these fixed representatives. The remarkable thing here is the evolution algorithm that delivers whatever the situation calls for. In the second simulation, Axelrod let the population matches with itself instead of the 8 representatives. The evolutionary force now has a moving target (which is itself). Naturally this would invoke the feeling of a fixed point theorem by John Nash.

Anyway, Axelrod runs the simulation too short (50 cycles) to see the cycle of cooperation and defection. He only sees the move from defective population to cooperative population.

The next paper on repeated game simulation that I would like to visit is by Fogel in 1993: Evolving behaviors in the iterated prisoners dilemma. Fogel did one thing differently, he uses the

finite state machine to represent strategies, maximum of 8 states in a machine. Here is the setup of his work:
- The population has 100 agents.
- Each agent matches with every other agent (round robin) for a game of 151 rounds.
- Each trial runs for 200 cycles.
- Multiple trials are run (total 20).

Because he runs for only 200 cycles, he sees the rise of mutual cooperation from randomness but the simulation is not long enough to see the collapse of mutual cooperation into mutual defection and the cooperation coming back again in the society.

Also, in the result of his simulation, there is a machine that alternates between cooperating and defecting. This is due to the fact that he sets the payoff $5 + 1 = 2 * 3$. Usually the temptation payoff plus the punishment payoff is smaller than two times the reward payoff.

So far the work I have presented make use of computer simulation. I, therefore, would like to present some more work about population game that approach from another angle: theoretical calculation. Of all the work in the literature, I would like to visit Rubinstein (1986) - Finite automata play the repeated prisoners dilemma and Binmore (1992) - Evolutionary stability in repeated game played by finite automata. In 1986, Rubinstein writes a paper on finite automata playing the repeated prisoners dilemma. However, he does not run simulation on it. He only does theoretical calculation on the equilibrium set. The main concern in his work is about the complexity cost of the automaton and how it affects the equilibrium selection process. In short, he works on the situation of a hyper rational person that delegates the strategy to a bounded rationality agent. For example, if 2 automata achieve the same result, the one with less states is the winner. This makes it systematically difficult for tit for tat when compared with always-cooperate. Because tit for tat has 2 states, while always-cooperate has only 1 state, and they are meant to deliver the same cooperative result. In 1992, Binmore considers the hyper rational person to be the evolutionary force of natural selection, and continues to elaborate on Rubinsteins work. He incorporates the condition of minimizing the number of states of an automaton in to the equilibrium concept called Modified ESS.

Continue with the thread of computer simulation, Nowak and co. ran a series of extensive simulations that I would describe next. In 1992, Nowak runs a simulation on the repeated prisoners dilemma on spatial network. He considers only the deterministic all-cooperate and all-defect automata. Since the simulation is run on a square, it results in beautifully patterned squares. The work is published as Nowak and May (1992) - Evolutionary games and spatial chaos. In 1992, Nowak and Sigmund published more work on Tit for tat in heterogeneous populations. They consider strategies with a Markov chain structure. The agent structure, however, is limited to reactive strategies. One strategy has only two pieces of information: the propensity to cooperate after opponent playing C and the propensity to cooperate after opponent playing D. For example, a tit for tat written in this way is (1, 0) or a stochastic tit for tat is (0.99, 0.01). The all-cooperate one is (1, 1). The all-defect one is (0, 0). In 2005, Imhof, Fudenberg and Nowak published another contribution on Evolutionary cycles of cooperation and defection. In this paper, they consider only 3 strategies: all-cooperate, all defect, and tit for tat in which tit for tat suffers a small complexity cost. They plot the replicator dynamics of this limited set of strategies and the population cycles around with different patterns due to calibrations.

In 1993, Nowak and Sigmund published another important contribution on A strategy of Win stay, lose shift that outperforms tit for tat in the prisoners dilemma game. This is one of the papers that deal with the strategy space in a broader way. Here the strategy has a Markov chain structure. The strategy remembers the outcome of the previous round and prescribes the tendency to cooperate in the current round. Because there are 4 possible outcomes (CC, CD, DC, DD) the strategy has 4 pieces of information. For example: tit for tat is represented as (1, 0, 1, 0). Here is an example of the stochastic tit for tat (0.99, 0.05, 0.98, 0.02).

With their simulation setting, they find out that the population cycles around defection and cooperation but in far apart timing. The second finding is that the mechanism that sustains cooperation is tit for tat sometimes but most of the time it is a strategy called win-stay lose-shift (1, 0, 0, 1). If you are winning then continue doing whatever you are doing, if you lose then switch action.

In 2013, Hilbe, Nowak and Sigmund wrote a paper on Evolution of extortion in iterated prisoners dilemma games In response to: Press and Dyson (2013) - Iterated Prisoners Dilemma contains strategies that dominate any evolutionary opponent. The story goes that in 2013, Press and Dyson calculate a strategy class called extortioner whose main purpose is to try to drive opponent in playing their way. For example, they enforce a unilateral claim to an unfair share of the rewards. Certainly it is costly initially, but the game lasts, hence eventually opponent accedes to the extortion. Hilbe and co. then run simulations on this class of strategies. They find out that in large population, this type of strategy gains no better advantage than others. However, in small population or two distinct population setting, extortionists are able to hold their ground. They then relate this result to the evolution between host species and their endosymbionts.

In 2012, van Veelen, Garcia, Rand and Nowak published a paper on Direct reciprocity in structured populations. This paper considers a very general approach in which the whole strategy space is *theoretically* taken into account. They do not consider a knowledge-based subset of the strategy space. The well known result is that the population cycles between cooperation and defection because of indirect invasion. For example, tit for tat achieves a cooperative state for the society by punishing all-defect. But tit for tat is not against all-cooperate hence they can share the population neutrally. Due to random drift, the percentage of all-cooperate can increase sufficiently for the population to be vulnerable toward invasion of all-defect. Hence the cycle starts. The main point of this paper is that they consider another dimension: the assertive mating. In random matching, the defective state is very long and cooperation is hardly visited. But as they increase the assortment rate, cooperation becomes more sustainable. The simulation use finite state machines representing strategies and continual probability to implement the repeated game (instead of discount factor).

My work, though relevant to other work above in the sense that I run simulation on population game, is about Nash demand game. I, therefore, would like to list some work on this game in the literature that I am aware of. For example, in 2000, Axtell, Epstein and Peyton Young published their work on The emergence of classes in a multi agent bargaining model. In this paper, Young and co. study the nash demand game. If the game is one shot and the agents are equal and the matching is random, then the equitable equilibrium (in which everybody claims half the pie) becomes very stable. They identify 3 states: the pure nash equiribrium of the nash demand game whose outcome is (5, 5), the mixed equilibrium in which exists aggressive claimants and submissive claimants, and the last state of fractious (i.e. pure chaos). They calculate the time of transitioning between equitable equilibrium to the state of unequitable equilibrium and the fractious state regarding the perturbation rate and the size of the population etc.

Then they proceed to another case in which agents have color tags and they remember the history of how a color agent plays. This is basically switching from the one shot game to the repeated game. Because agents remember and condition based on their memory, the path toward discriminatory shares opens wide up and the population struggles to maintain the equitable norm. It spends a lot of time in other states: the inequitable one and the fractious (wasteful) one. Another work in 2010, by Poza, Galan, Santos and Lopes Paredes: An agent based model of the nash demand game in regular lattices. This paper uses similar premises but extend the model a bit by structuring the population into regular lattices.

Thanks to pointers from referees, I would describe here some work by J. Halpern and co. on agent game and resource sharing. Firstly, Multiagent learning in large anonymous games by Ian A. Kash, Eric J. Friedman, and Joseph Y. Halpern (2011) . "It is shown that stage learning efficiently converges to Nash equilibria in large anonymous games if best-reply dynamics converge." The result of their work supports the methodology that I am going to use for this thesis. It supports the assumption about the validity of this body of research in the literature that I stated at the beginning of this thesis.

A second work from this scholar that I would like to cite is No justified complaints: On fair sharing of multiple resources by Danny Dolev, Dror G. Feitelson, Joseph Y. Halpern, Raz Kupferman, and Nati Linial (2012). While my work is about bargaing of one single resource, their work is on sharing of multiple ones. They define fair sharing in that context to be that no one can complain anything about the allocation and then proceed to prove that that kind of fair sharing always exist under broad circumstance. "The proof, which uses tools from the theory of ordinary differential equations, is constructive and provides a method to compute the allocations numerically."

Chapter 2

The game and simulation configuration

2.1 The game

The Nash Demand Game is the game of dividing a pie. It is a rudimentary example of bargaining behavior. There is a pie of 10 euro to be divided between two players. Each player can make 3 possible claims on the pie: I want a low/ medium/ high portion of the pie. This corresponds to claiming 2 / 5 / 8 euro out of 10. The claims are made simultaneously. If the sum of the claims are compatible (i.e. it is not bigger than the whole 10), each gets what they want. Otherwise, negotiation fails and no one gets anything. Here is the payoff matrix of the game:

NDG	Low	Medium	High
Low	2,2	2,5	*2,8*
Medium	5,2	*5,5*	0,0
High	*8,2*	0,0	0,0

Table 2.1: Nash Demand Game payoff matrix. The stars mark the Nash equilibria.

I list some real life examples of the bargaining behavior. It can be the situation between buyer and seller in the market, where they try to divide the distance between the willingness to buy and the willingness to sell. It can be the situation between employer and employee negotiating salary. It can be the situation of claiming natural resources of the earth among conventional nations.[1]

The one shot game has 3 pure Nash equilibria (marked in the table) and 1 mixed equilibrium of Low and High strategy. The three pure equilibria are, rationally speaking, equally viable. From the efficiency perspective, it does not matter how the pie is shared. As long as everything is utilised, it is considered a neat solution. However, the way we see it, from another angle, the equitable share (5-5) is symmetric and the non-equitable share (8-2 or 2-8) is asymmetric. That irritates a bit.

This simulation runs on two settings: one shot and repeated game. I list here some ways to interpret these settings. The one shot interaction can be considered as interaction between strangers. If so, the repeated interaction is the interaction between people who have enough data history to form expectation on how the other plays. In Axelrod's word, it is the game between opponents who have informative past and important future. Or simply they get stuck with each other due to circumstances. Another way to see it, the one shot game is the game between myopic players. So no matter how long it is played, they only care about immediate payoff such as of today and tomorrow maybe. Promises about stuff coming after tomorrow is considered noise. The repeated game, on the other hand, would be between patient players who feel like payoffs far in the future is also worth waiting for.[2]

[1]The brave interpretation from pure theoretical work to reality is, however, not recommended.
[2]The interpretation is a bit stretching, I believe.

2.2 The problem

The problem that the game theory simulation literature is going after is, in general, searching for optima in a vast space. The strategy space of one shot PD has 1 dimension. Let P_C be the probability of choosing C then the probability to choose D is $1 - P_C$. If we are dealing with a repeated game of 3 rounds, the number of strategies is $2 * 2 * 2 = 2^3 = 8$. The possible strategies are:

```
C - C - C  ->  CCC
  \   \ D  ->  CCD
    D - C  ->  CDC
      \ D  ->  CDD
D - C - C  ->  DCC
  \   \ D  ->  DCD
    D - C  ->  DDC
      \ D  ->  DDD
```

The strategy space of this game, hence, has 7 dimension. If we suppose to play the game infinitely, the number of strategies is 2^∞, the number of dimensions of the strategy space, therefore, is $2^\infty - 1$. It is incredible hard to find the optima in such a vast space, therefore, I approach by doing simulation.

2.3 Evolutionary algorithm

Population and Agents Starting from a world with many slots, we populate them with agents. The population will be kept fixed and each agent will adopt a strategy/machine to execute when playing the game. A brief definition of agents is that agents do agency: given a strategy, they play by the book until they terminate.

Cycle A typical cycle has 3 phases: matching phase, learning (or regeneration) phase, and mutation phase. In the matching phase (Figure 4.1), agents will be matched randomly in pair to play the game for multiple rounds. Their payoff sequence will be collected to calculate the corresponding fitness (i.e. the relative share of each final payoff in the total population payoff).

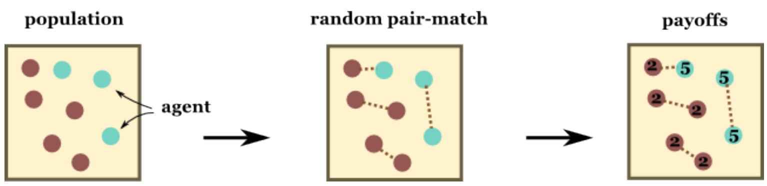

Figure 2.1: Matching phase

The fitness vector of the whole population will be used in the regeneration phase (Figure 4.2). In the regeneration phase, a small number of agents will be allowed to *learn*. Technically, they randomize over the fitness vector to choose their new strategy. Because the fitness of each strategy is different, a better strategy will be more attractive and it'll be more likely to be chosen hence it'll become more popular at the next cycle. In this way, the strategy that performs better will grow at the expense of the poor doers.

In the mutation phase (Figure 4.3), a small number of agents will experiment on their current strategy (mutating). This is to keep adding variety into the selection pool.

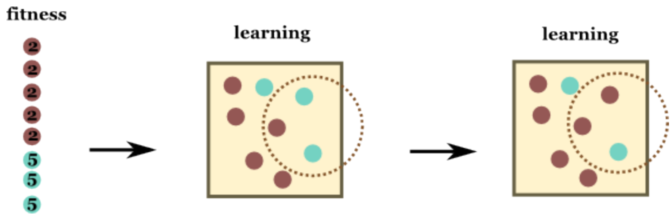

Figure 2.2: Learning phase

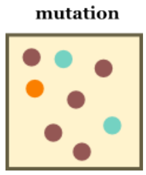

Figure 2.3: Mutation phase

Rationale Suppose there is X strategies available in the strategy space. If we start the simulation with a population of X agents holding all possible strategies of the game, the learning process would be sufficient to single out the fittest survivors given that we run the simulation long enough. To accommodate for the likely possibility of multiple equilibria, we run the simulation many many times to collect the set of different survivors. The winner can be a pure strategy or it can be the population state of a mixed strategy. X, however, is infinite. What we can do is that we start the simulation with an initial population of N = 100 agents. If we do not use the mutation process, the selection process would choose the best strategy among the 100 strategies used by these agents. The initial population therefore affects the ending result. The mutation process is to keep adding varieties into the selection pool. By doing this, the initial population does not matter. Because the population would keep receiving new strategies over cycles, we can gradually cover a good portion of the strategy space.

2.4 Payoff calculation

Since a match of 500 rounds gives a sequence of 500 payoffs, I use factor δ as the bank uses to calculate the present value of this future-involved sequence. This is a repeated game and we interpret the sequence from round 2 onwards to be in the future. A myopic person has a small δ which means that how much she considers to get from the 500 round game relies heavily on the first few rounds. After that, even a series of highest possible payoffs would not interest her. She considers high payoffs in the future to be 0 and dismiss them. This kind of person would prefer to get high payoffs in the first few rounds, she would choose the sequence 2 over sequence 1 in the following example:

```
sequence 1: 0 0 0 0 4 4 4 4 4 4 4...
sequence 2: 4 4 4 0 0 0 0 0 0 0 0...
```

On the other end of the spectrum, a person with high δ is a highly patient person. She can wait for the payoffs in the future or at least she would take them into account. For her, 1 euro of tomorrow worths almost as 1 euro of today. She would prefer sequence 1 over sequence 2 with sufficiently high δ.

There is another interpretation of δ as continual probability in which the first round is played for sure. At the end of that, a δ biased coin is flipped, with probability δ the game continues, and with probability $1 - \delta$ the game stops.

Mathematically these two interpretation converges but in implementation, regarding the same δ, it is more stable to implement it as discounting factor, which I do here.

I can also run the simulation with the continual probability implementation, the result, though qualitatively similar, is very noisy making it difficult to see the point.

The noise comes from the fact that, for 50 pairs of even the same agent, the numbers of rounds varies greatly even though the continual probability δ come from the same distribution. For example, let's say we have 10 agents using the same strategy. Since δ is an independent and random draw, these 10 agents can get to play 10, 100 or 150 rounds which give a volatile payoff despite the fact that they are using the same strategy. Of course, with the law of large numbers, the average payoff should reflect well the merit of the strategy but here we are dealing with small numbers. This volatility unnecessarily interfere with the calculation of the fitness of a strategy, in my opinion.

2.5 500 rounds

I usually set the number of rounds to be 500. Because at the maximum δ = 0.99, the payoff at round 500 would be negligible. I assume that the match can be safely cut off after this threshold. In other words, I can safely assume this to be an infinite match. Certainly one can ask about δ = 0.9999 but I would say that rounding up to 2 decimals is a reasonable assumption about how the world works. Also, in computing, these imprecise approximations are common (see: floating numbers). They serve reality well.

2.6 Strategy implementation: finite state machine

Here is an example of a strategy represented by a finite state machine: the one that always cooperates.

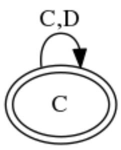

Figure 2.4: The cooperator

This machine has one state. The action prescribed in that state is to cooperate. And the arrow with letter C and D means that if the opponent acts C or D in the current round, in the next round this machine stays in the same state of Cooperating. The double circle means that this is the initial state to start with. To illustrate it better, here is the example of another classic strategy: Tit for Tat:

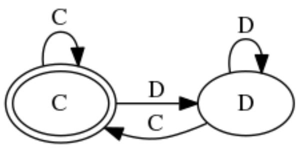

Figure 2.5: Tit for Tat

This machine has 2 states. The double circled state which contains letter C is the initial one. Hence this machine starts out cooperating. In the current round if the opponent C, in the next round the machine stays in state C. However if the opponent D, in the next round the machine jumps to state D.

Here is the Grim Trigger:

The Grim Trigger starts out by cooperating which is optimistic. As long as the opponent plays C, it stays in the state of Cooperating. Once the opponent switches to D, it jumps to D and never looks back. This one does not forgive and does not forget.

The mutation can change initial state, the letter in a circle, the ending point of an arrow or it can add, detach an entire circle.

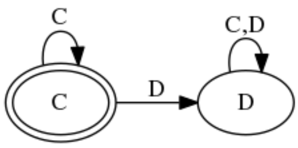

Figure 2.6: Grim Trigger

2.6.1 More examples

Since in this section I run the simulation with the Nash Demand Game, here are some examples of the relevant machines: the one who always claims Low, Medium or High.

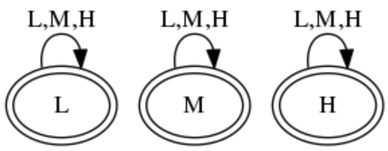

Figure 2.7: Always-low, always-medium, always-high

This one is an Flexible machine: it starts out playing Medium and chooses what to do next based on how the opponent plays in the previous round. If you play H it retreats to L, if you play M it play M, if you play L it claims H.

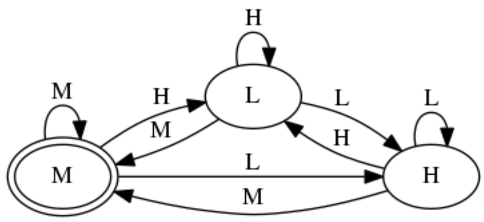

Figure 2.8: Flexible machine

Here is the payoff sequence of a match between always-low and always-medium:

always-low	always-medium	always-low	always-medium
L	M	2	5
L	M	2	5
L	M	2	5
..	..		

Table 2.2: Always-low and always-medium

between always-medium and always-high:

always-medium	always-high	always-medium	always-high
M	H	0	0
M	H	0	0
M	H	0	0
..	..		

Table 2.3: Always-medium and always-high

To get a glimpse of the Flexible machine character, let's match it with the always-low, always-medium, always-high and itself:

always-low	Flexible machine	always-low	Flexible machine
L	M	2	5
L	H	2	8
L	H	2	8
..	..		

Table 2.4: Always-low and Flexible machine

always-medium	Flexible machine		
M	M	5	5
M	M	5	5
M	M	5	5
..	..		

Table 2.5: Always-medium and Flexible machine

always-high	Flexible machine		
H	M	0	0
H	L	8	2
H	L	8	2
..	..		

Table 2.6: Always-high and Flexible machine

Flexible machine	Flexible machine		
M	M	5	5
M	M	5	5
M	M	5	5
..	..		

Table 2.7: Flexible machine and itself

We can see that, in the match with always-low, the Flexible machine switches to play High from the second round. In the match with always-high, the Flexible machine switches to play Low from the second round. These are not to waste the resources.

2.7 How to interpret the result

When the number of states of a machine is small, it is easier to lay out its scheme and describe it in a comprehensible and sensible way. When the number of states is really big, there are so many contingencies it makes the plan becomes incomprehensible. It makes better sense seeing what the machine does instead of looking at its algorithm and making guesses. Which means that we would match it with some classic machines and look at its payoff sequence. The intuition is that if we cannot tell who you are, we watch your interaction with other known members and we make deduction about your supposedly revealed nature. [3]

The benchmark is the set of classic machines that the simple character has already been established. They are the always-lows, always-medium, always-high and the Flexible machine.

Taking the number of rounds to be 500, and the δ to be 0.99.

There are 3 pure Nash equilibria which are marked with stars in the table. Now if we add a random machine into this benchmark matrix, it may or may not change the status of the equilibrium points. Let's take an example that shakes up the status quo:

In this table, we can see that machine a forms an equilibrium with itself. And in this equilibrium, the payoff average (which is 322.9) is less than the case of an always-medium equilibrium

[3] This sounds like a dangerous procedure. Applying on sophisticated beings is strongly not recommended.

0.99	always-low	always-medium	always-high	Flexible machine
always-low	199 199	199 497	*199 795*	199 792
always-medium	497 199	*497 497*	0 0	497 497
always-high	*795 199*	0 0	0 0	787 197
Flexible machine	792 199	497 497	197 787	497 497

Table 2.8: Payoff benchmark, and equilibria

0.99	machine a	always-low	always-medium	always-high	Flexible machine
machine a	*322.9 322.9*	670.8 198.7	267.7 267.7	80.6 322.5	532.5 297.2
always-low	198.7 670.8	198.7 198.7	198.7 496.7	*198.7 794.7*	198.7 791.7
always-medium	267.7 267.7	496.7 198.7	*496.7 496.7*	0 0	496.7 496.7*
always-high	322.5 80.6	*794.7 198.7*	0 0	0 0	*786.7 196.7
Flexible machine	297.2 532.5	791.7 198.7	*496.7 496.7	196.7 786.7*	496.7 496.7

Table 2.9: The stability of a random strategy

(which is 496.7). This is a strategy that sustain wasteful negotiation in the society. It is able to form a stable equilibrium with itself usually because it is able to resist always-high. If it submits to always-high or Flexible machine, the star would go to the always-high or Flexible machine. In other word, always-high or Flexible machine would be a better response to this strategy than it is to itself.

2.7.1 Evolutionarily stable strategy

The central concept in evolutionary game theory is about an evolutionarily stable strategy, proposed by Maynard Smith [29]. This concept is used to describe the persistence of one strategy against another strategy, illuminating in a population context. The scenario is as following: there is one population hosting exclusively strategy A. A can be a pure or a mixed strategy. When there is ϵ mutation B entering the population, A is said to be evolutionarily stable against B if A can repel B. In the other way around, B invades the population of A. On the middle ground, they are neutrally stable which means that they coexist at any possible ratios.

For example, in the one shot prisoner's dilemma, there are only two strategies: to play C and to play D. Let's say there is a population of people playing D, hence strategy D is an incumbent. By mistake, some people switch to play C, hence strategy C is a mutant.

PD	C	D
C	3	0
D	4	1

Table 2.10: In the one shot prisoner's dilemma, D is evolutionarily stable against C

To answer the question, whether D is evolutionarily stable against mutant C, we consider two criteria:

- First, when being matched with the incumbent D, an incumbent D gets 1 while a mutant C gets 0. $1 > 0$ hence the weakest definition of rationality prescribes to prefer D.

- Second, when being matched with the mutant C, an incumbent D gets 4 while a mutant C gets 3. $4 > 3$ hence the incumbent D is still doing better.

The incumbent is doing better than the mutant in both cases: matched with the incumbent and matched with the mutant. Hence the mutant is unable to spread in the population and the incumbent is stable.

Consider the example in the repeated Nash Demand game:

The stars indicate two pure Nash equilibria. However, in the context of evolutionarily stable concept, to answer the question, whether the incumbent a is stable against the invasion of the mutant always-medium, we consider two criteria:

- First, when being matched with the incumbent a, a machine a gets 323 while a mutant always-medium gets 268. $323 > 268$ hence the incumbent is already doing better than the mutant. The

0.99	machine a	always-medium
machine a	*322.9 322.9*	267.7 267.7
always-medium	267.7 267.7	*496.7 496.7*

Table 2.11: Machine a is stable against always-medium.

mutant cannot spread. We do not need the second criteria because the incumbent is already the best response to itself.

- Second, when being matched with the mutant always-medium, a machine a gets 268 and a mutant always-medium gets 497. $268 < 497$ this is to say that the mutant is also a best response to itself and it would be stable against this machine a if it is the incumbent.

In the case above machine a is able to resist the invasion of always-medium because machine a is not kind to always-medium. It gives always-medium not enough point. Let's consider another example where machine a gives always-medium enough point to destabilise itself:

0.99	machine a	always-medium
machine a	322.9 322.9	267.7 400*
always-medium	*400 267.7	*496.7 496.7*

Table 2.12: Example continued: machine a is no longer stable.

When being matched with the incumbent a, a machine a gets 323 while a mutant always-medium gets 400. $323 < 400$ hence always-medium would easily take over the population. Machine a is no longer a best response to itself because it gives too much payoff to the mutant always-medium. Machine a is no longer a Nash equilibrium, it has lost all the stars.

0.99	machine a	always-medium
machine a	*322.9 322.9*	267.7 322.9*
always-medium	*322.9 267.7	*496.7 496.7*

Table 2.13: Example continued: machine a is not evolutionarily stable.

In this case, when being matched with the incumbent, an incumbent a gets 322.9 and a mutant always-medium gets the same point 322.9. So they are equally fit. Here is where we should consider the second criterion: when being matched with the mutant, an incumbent a gets 267.7 while a mutant always-medium gets 496.7. The mutant is able to get better for themselves and the mutant does not treat the outsider kindly. Hence in this case machine a is not an ESS, even though it is still a Nash equilibrium of the game. The evolutionarily stable concept is then considered a stringent (refinement) of the Nash equilibrium concept.

If they get equal payoff everywhere they are neutrally stable and they can coexist at any possible rate. The population can drift between the two for decades.

That's for the evolutionarily stable concept for the interpretation of my simulation result. We can see that, in a 3x3 game, the strategies become so much more complex and so much more dimensional. An example is that one machine can have 100 states and this 100 states can generate explosive numbers of contingency. It is impossible to interpret all these contingencies in a comprehensible manner. I have given up on imposing a scheme of personality test to evaluate the characteristic of the all strategies. I use certain indications but I remain to proceed case by case.

Chapter 3

Simulation result: the repeated Nash Demand Game

3.1 Patient players

I run with the following configurations: the population has a fixed N agents and N = 100. The number of rounds to be played when two agents are matched is 500. The delta to calculate present value of the 500 payoff sequence is 0.99. The learning rate is 10 percent. The mutation rate is 2 percent.

$\delta = 0.99$ is very close to 1. This represents a highly patient agent. Then I plot the average payoff of the whole population, with 3 benchmarks. The red line is the hypothetical payoff resulting from a sequence of highest possible payoff: 8. The blue line is the hypothetical payoff resulting from a sequence of all payoffs of 2. The green line is when there is 5 everywhere (i.e. everyone is claiming 5 all the time). I increase the mutation rate because the strategy space of a 3x3 game is exponentially bigger than the strategy space of a 2x2 game.

3.1.1 $\delta = 0.9$

Simulation 1

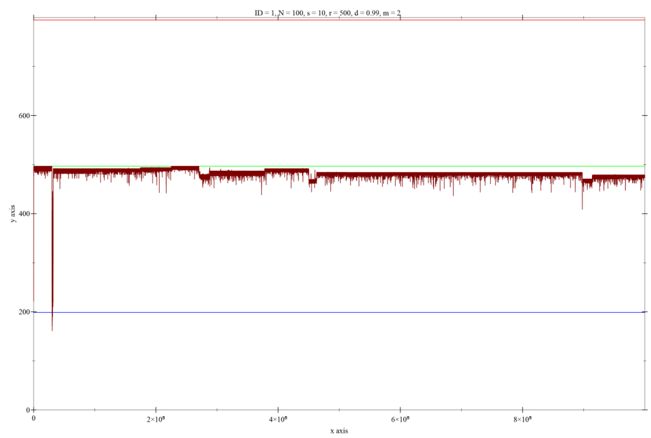

Figure 3.1: One typical run of 1 000 000 cycles: $\delta = 0.99$

Overall I would like to make some brief comments. The first one is that there is very little time that the 50-50 division holds. However, in these non-equitable periods, the population average payoff does not goes down too low. When I look into the population, most of the reason is that agents are using strategies that waste resources for several rounds. This kind of probing behavior does not necessarily happen in the initial period. It can happen after the initial period. The common thing is that this limitedly wasteful period reduces the population average. It becomes less than ideal but not much. Outside the wasteful period, they all play medium. The characterisation of the wasteful period is some show of strength, some probing and some retaliation. The third comments is that there are periods that the population average is extremely low and chaotic. This one is due to a mixture of truly aggressive strategy and its submission (aggressor and Flexible machine). This one can legitimately be a stable mixture. Lastly, there are periods that the population average is not perfect due to a different reason. It can be due to a bad luck in mutation and the state is unstable waiting for the right mutation to appear.

For further information, I plot the first 50K cycle of Figure 7.1. This short period includes the stable 50-50 division, its collapse and raise up again. Hence we would be able to watch in action the series of events leading to its fall and see in details what helps to pave its way back. It comes back less than ideal a little (due to an initial period of probing), but would be back to full 50-50 division at around cycle 200000.

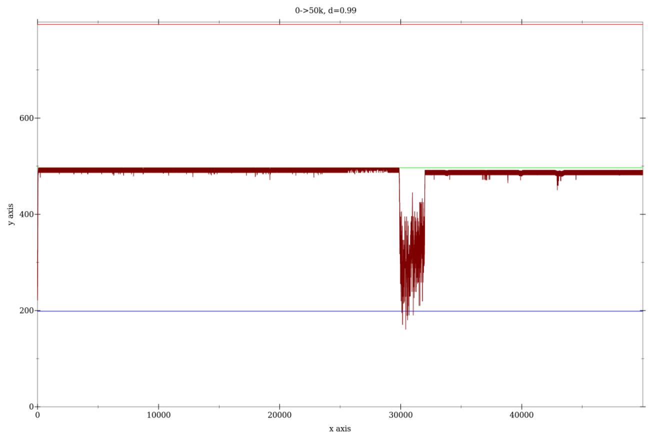

Figure 3.2: $\delta = 0.99$, cycle 1 -¿ 50K

Point 1 Until cycle 25000 (or precisely cycle 24800), the population is full of all-medium strategies. The description of an always-medium strategy is that it plays medium to itself, to always-low, to the Flexible machine and it gives always-high zero payoff. Because it gives always-high nothing, it is able to resist always-high invasion.

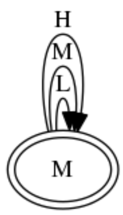

Figure 3.3: all-medium strategy, at cycle 25K

0.99	always-low	always-medium	always-high	Flexible machine
always-medium	496.7 198.7	*496.7 496.7*	0 0	496.7 496.7*

Table 3.1: From cycle 0 to cycle 24800: always medium

We would see the evolution of the strategy from cycle 24800 to cycle 30K which is the transition from perfectly equitable strategy to a chaotic collapse.

From cycle 24800 to cycle 25800, the strategy is no longer the always-medium because it has mutated a new feature: it gradually submits to always-high strategy. The always-medium strategy always play medium with itself, with the always-low and the Flexible machine. And it resists the always-high completely, i.e. it does not gives always-high a single point because it never submits to the aggressive claim of always-high. It insists on claiming 5 while always-high insists on claiming 8. Because of this, always-medium is a best response to itself. It is able to form an equilibrium with itself. From cycle 24800 to cycle 25800, the machine starts to submit to always-high, i.e. it claims 2 and always-high starts to get a lot of points. The behaviors with other classic strategies do not change yet, for example, it still claim 5 toward itself, always-low, and Flexible machine. But it starts to submit if the opponent claims aggressively. In this period, the always-medium equilibrium is still stable because the point it gives to always-high is not high enough. Always-high is still not a best response and the strategy is still a best response to itself. This situation would change soon.

0.99	always-low	always-medium	always-high	Flexible machine
the compromised always-medium	496.7 198.7	*496.7 496.7*	57 228	496.7 496.7*

Table 3.2: From 24800 to 25800, cycle 25100: always-medium is compromising toward always-high. Not enough to destabilise itself, but soon.

Point 2 At cycle 25900, the machine is no longer a stable equilibrium. It has gives always-high enough point to destabilise the 50-50 division. Always-high has become a best response to the current strategy of the population.

0.99	always-low	always-medium	always-high	Flexible machine
the compromised always-medium	496.7 198.7	496.7 496.7	175 700*	496.7 496.7

Table 3.3: Cycle 25900: the compromised strategy. It can no longer be called an always-medium, even though it still plays fair with itself, always-low, and Flexible machine.

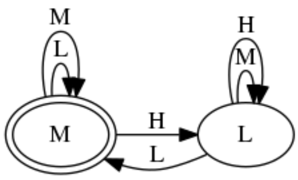

Figure 3.4: At cycle 26K, the strategy evolves to accommodate aggressive opponent. But the situation is still fine because no one happens to be aggressive yet. It is a Compromised strategy. On face value, it happens to generate modesty exactly as it is supposed to do, but it has a branch that reaches out to all-high invasion.

The reason is that since 25900, the machine mutates another state with a new contingency (which is a security breach): if the opponent plays High, it switches to the newly added state of playing Low instead of insisting on playing Medium. This happens off equilibrium path. It goes silent because in the society people keep claiming Medium to each other and the population payoff average is still ideal all the time. No one happens to claim High in this environment hence no one ever knows that the other would retreat immediately.

Here I make a less relevant remark. Binmore makes a fortified concept of the ESS: the modified evolutionarily stable strategy. It is due to the idea in Rubinstein's paper that: in a fortified equilibrium, the machine drops the unnecessarily state. Hence the machine in figure 7.4 would not be considered a MESS because it has a state of playing Low that is never reached in the equilibrium scenario of the 50-50 division.

Anyway, back to my simulation. From this point on, the strategy opens itself to the possibility of submitting to aggressive negotiator, however, it does not itself consider that possibility yet. As I show in the table later, it does not learn to exploit the Flexible machine or the always-low, one that submits immediately if you go bold.

Overall in this period, the population average payoff is still fully equitable. Everybody is still sharing 50-50 all the time. But the sustaining mechanism is not the same as the mechanism provided by the always-medium. The difference is that always-medium is able to resists always-high. This strategy a is modest to itself all the time but it submits to always-high. It plays low all the time in response to always-high. This one is neutral to always-medium and by some random drift its percentage in the population can grow up very fast.

This neutral mutant makes the population weak because it is vulnerable to all-high strategy. It submits completely to always-high hence it is unable to maintain an equilibrium with itself. It is unstable. As I show in the table, it still generates equitable shares among itself but the fact that it submits to all-high makes it no longer best response to itself. all-high becomes the best response to this machine. It is just that in this situation, everybody is claiming Medium all the time and no one happens to discover that they can play aggressively all the time yet.

0.99	machine a	always-low	always-medium	always-high	Flexible machine
machine a	496.7 496.7	496.7 198.7	*496.7 496.7	196.7 786.7*	496.7 496.7
always-low	198.7 496.7	198.7 198.7	198.7 496.7	*198.7 794.7*	198.7 791.7
always-medium	496.7 496.7*	496.7 198.7	*496.7 496.7*	0 0	496.7 496.7*
always-high	*786.7 196.7	*794.7 198.7*	0 0	0 0	*786.7 196.7
Flexible machine	496.7 496.7	791.7 198.7	*496.7 496.7	196.7 786.7*	496.7 496.7

Table 3.4: Machine a at cycle 26k. It no longer is a best response to itself, which is a security breach, because it opens to the possibility of accommodating aggressive claims. It does not contemplate itself on playing aggressive though.

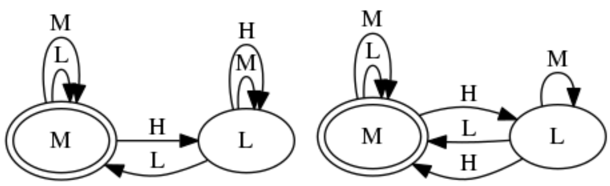

Figure 3.5: At cycle 27K, 85% machine a submitting to all-high completely and 15% machine b submitting to all-high half the time.

Point 3 In cycle 27K, the mutated machine b actually can handle all-high strategy. It submits to all-high only half the time hence able to resist the invasion of all-high strategy.

0.99	machine b	always-low	always-medium	always-high	Flexible machine
machine b	*496.7 496.7*	496.7 198.7	*496.7 496.7	99 395	496.7 496.7
always-low	198.7 496.7	198.7 198.7	198.7 496.7	*198.7 794.7*	198.7 791.7
always-medium	496.7 496.7*	496.7 198.7	*496.7 496.7*	0 0	496.7 496.7*
always-high	395 99	*794.7 198.7*	0 0	0 0	*786.7 196.7
Flexible machine	496.7 496.7	791.7 198.7	*496.7 496.7	196.7 786.7*	496.7 496.7

Table 3.5: Machine b at cycle 27k. It is best response to itself and has all the stars. This one is an always-medium strategy. The compromise is not serious yet. It is still able to resist always-high. Machine a, on the other hand, cannot.

If machine b can grow up in numbers, the population can be stable again, the security breach is closed. However, because there is no force to push against machine a or machine b, the percentages of them are just due to random drift. They both get the same payoffs, they play medium among themselves and to each other. They coexist peacefully together. It is just that beneath the surface, one is secretly compromised, one is not.

At cycle 28k and cyle 29k, fate takes a turn and the evolution drifts toward the weak one. The strategies become vulnerable to all-high.

Point 4 At cycle 30000, the 50-50 division norm collapses. The population is a mixture of 3 machines.

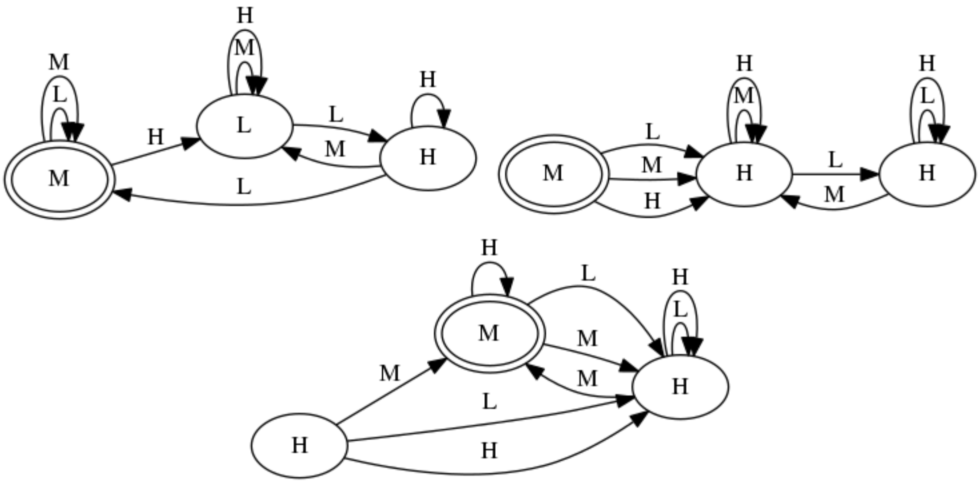

Figure 3.6: At cycle 30K, 41% machine c, 34% machine d, 25% machine e.

I would describe here the machine and the payoff sequence they get.

Machine c is very moderate to itself, but it submits quickly to machine d and machine e. This explains the chaos in the population payoff average. There are two sequences of 5s (5 5 5 5...) for the match of machine c with itself. But there would be a sequence of 2s (2 2 2 2...) and a sequence of 8s (8 8 8 8...) for the match between machine c and machine d or e. The match between machine d and e creates two sequences of 0s (0 0 0 0...) because they do not claim compatible shares.

If you have a look at machine c, you would see that it starts by playing Medium. If opponent plays High it plays Low, and it is able to switch to play High if opponent plays Low. This machine has evolved to be able to exploit. The previous ones learn to submit to all-high but this one has gone one step further, it has learned to be flexible enough and play aggressive. It has a touch of an Flexible machine.

Please have a look at machine d, it plays Medium for only the first round. Then it has two extra states but both claiming High. This one is almost the all-high strategy.

Machine e also looks complicated but it has only 2 states. There is a redundant state that there is no way to reach it from anywhere inside the machine.

Because machine d and e does not have a state of playing Low, they play it aggressively to each other all the time, they both get 0 if being matched. Machine c is to cushion the aggression because it has a state of playing Low.

Overall, we can treat this mixture as one. This mixture is hostile toward always-high because it has enough aggressive machines in the mixture. It also learns to exploit always-low and Flexible machine. The best response to this kind of strategy is the Flexible machine. This kind of situation (a mixture between aggressor and Flexible machine) happens from 29900 to 31900.

Point 5 At cycle 31K, the mixture evolves better: 58% machine e and 42% machine f. Machine e is an all-high machine. Machine f is almost an Flexible machine, it is flexible.

0.99	mixture 1	always-low	always-medium	always-high	Flexible machine
mixture 1	*322.9 322.9*	670.8 198.7	267.7 267.7	80.6 322.5	532.5 297.2
always-low	199 671	198.7 198.7	198.7 496.7	*198.7 794.7*	198.7 791.7
always-medium	268 268	496.7 198.7	*496.7 496.7*	0 0	496.7 496.7*
always-high	322 87	*794.7 198.7*	0 0	0 0	*786.7 196.7
Flexible machine	297 532	791.7 198.7	*496.7 496.7	196.7 786.7*	496.7 496.7

Table 3.6: The mixture at cycle 30k. So much resource wasted because of the match between aggressive claimaints. Note that this mixture has become slightly aggressive to always-low and the Flexible machine.

This explains why they maintain the mixture of half each. Machine f submits to machine e. Machine f is moderate with itself. Machine e is zero with itself.

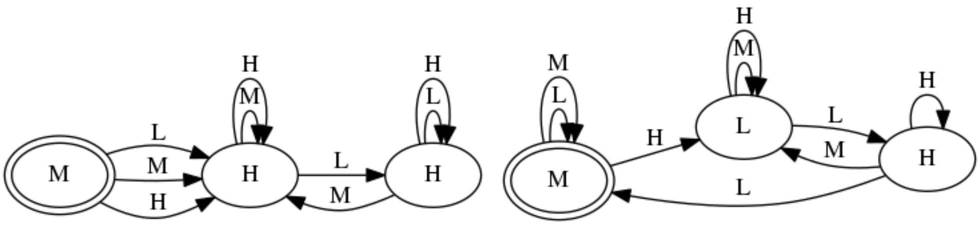

Figure 3.7: At cycle 31K, 58% machine e, 42% machine f.

If you look at the character of machine e, it only has a state of playing High. It is an all-high strategy. Machine f has 3 states of Low, Medium and High. This mixture is not overall stable though. The population average quickly raises up after 1000 cycle.

0.99	mixture 2	always-low	always-medium	always-high	Flexible machine
mixture 2	328.9 328.9	667.8 198.7	211.5 211.5	82.6 330.4*	663.2 324.4
always-low	199 678	198.7 198.7	198.7 496.7	*198.7 794.7*	198.7 791.7
always-medium	212 212	496.7 198.7	*496.7 496.7*	0 0	496.7 496.7*
always-high	*330 83	*794.7 198.7*	0 0	0 0	*786.7 196.7
Flexible machine	324 663	791.7 198.7	*496.7 496.7	196.7 786.7*	496.7 496.7

Table 3.7: The mixture at cycle 31k. It is no longer stable because it gives always-high more than it gives to its own kind.

Point 6 At 31900, there appears a new pure strategy that is able to be stable. It takes over the stable mixture of aggressor and Flexible machine. It shares certain features with the mixture: it exploits the weaks (always-low and Flexible machine) and it resists the aggressors (always-high), it resists the always-medium. But most importantly, the sufficient condition is that it plays good enough with itself when it recognises its own kind hence it is able to form an equilibrium with itself. This is a remarkable strategy. It is a tough strategy.

0.99	tough	always-low	always-medium	always-high	Flexible machine
tough	492 492*	792 199	5 5	0 0	784 200

Table 3.8: Starting at 32100, a tough kind.

From 32000, the population payoff average comes right back to the ideal level, a little less. The strategy is to resist always-high completely and be kind enough to its own kind. This machine recognises its own kind by probing in the first round. The payoff sequence hence 0 5 5 5 5 ... It plays aggressively in the first round but then retreat to play Medium onwards. It does not retreat with always-high. It does not be kind to always-Medium. It even recognises to exploit always-low and Flexible machine. I think that this one is a remarkable strategy. It has the ability to recognise one's own kind to play it good with itself. It repels always-high, always-medium and exploits always-low and forces Flexible machine to submit to it.

I describe again. The Flexible machine would be able to play fair with itself, exploit all-low, submit to all-high. This one, with an extra state, is able to distinguish between an Flexible machine and its own kind. It exploits Flexible machine while still be kind to itself. Plus, it resists always-high and always-medium. It forms an equilibrium with itself in the presence of all formidable strategies. It is a tough player.

0.99	tough	always-low	always-medium	always-high	Flexible machine
tough	*491.8 491.8*	791.7 198.7	5 5	0 0	783.8 199.7
always-low	198.7 791.7	198.7 198.7	198.7 496.7	*198.7 794.7*	198.7 791.7
always-medium	5 5	496.7 198.7	*496.7 496.7*	0 0	496.7 496.7*
always-high	0 0	*794.7 198.7*	0 0	0 0	*786.7 196.7
Flexible machine	199.7 783.8	791.7 198.7	*496.7 496.7	196.7 786.7*	496.7 496.7

Table 3.9: Cycle 32100 The tough strategy matching with others

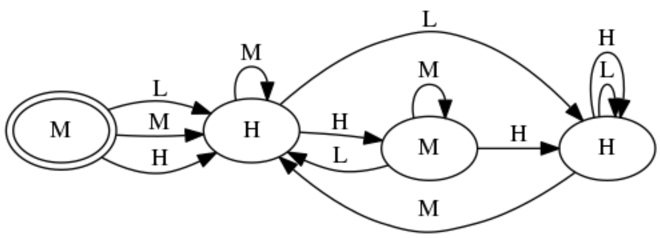

Figure 3.8: At cycle 32000, a tough negotiator. Tough 1

Actually, at cycle 32000, the percentage of this machine is 95% and it raises to 100% quickly after that.

The situation is stable until cycle 43000 where appears another mechanism. It is another kind of tough strategy, though. It probes in the second round hence the payoff sequence 5 0 5 5 5 5... It is able to distinguish the previous kind of tough with itself, hence it gives itself 492 point while it gives the previous tough machine less, 486. Funny thing is that the previous tough would give itself 492 but gives this one 486. They both use the initial rounds to probe and recognise themselves. This is the defining feature of a tough machine.

0.99	tough	always-low	always-medium	always-high	Flexible machine
tough	492 492*	795 199	0 0	0 0	786 197

Table 3.10: At 43500, another tough kind.

This kind of tough machine takes reign for an incredibly long period, until cycle 99800. Where a mutation happens that starts to reach out to always-medium. From cycle 99800 to 100800, the mutation gives always-medium some point though not enough to destabilise the status quo. At the same time of 100800, another mutation takes place that starts to submit to always-high and exploits Flexible machine and always-low less. The mutation is make the strategy less aggressive. It starts to play fair with itself, the always-medium, and the Flexible machine. It plays Low with always-low and gives always-high some more point. All these mutations happening it is still stable because it still resists always-high. It can be said to be some kind of always-medium[1].

From 106000 to 117100, the strategy is almost always-medium and still stable, but it grows to exploits always-low. After that, the mutation goes away silently. The strategy is to play fair with

[1]gone wild

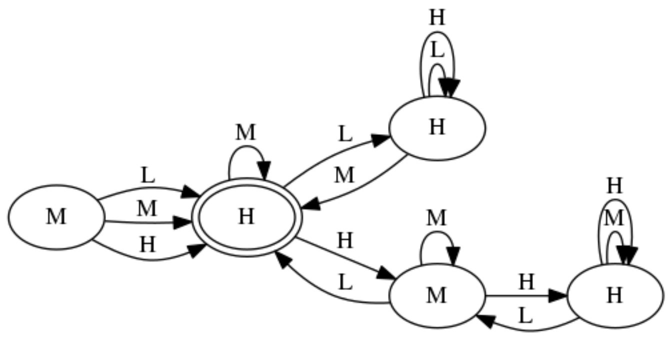

Figure 3.9: At cycle 43500, another tough negotiator. Tough 2

itself, always-medium, Flexible machine. It plays low with always-low and gives always-high some point. From 123900, it gives always-high 0 and is not kind to Flexible machine but the stability remains. From 138400, it starts to give always-high some point again.

From 147900, the population is a mixture of two machines. One machine (84%) submits to the Flexible machine completely and the Flexible machine becomes the best response. It still plays fair to itself and the always-medium. It plays Low to always-low and gives always-high some point. But it gives the Flexible machine a whole lot of points and it is no longer stable. The other machine (16%) is still able to form an equilibrium with itself. However, overall, as a mixture, the population submits to Flexible machine and is not stable.

0.99	x	always-low	always-medium	always-high	Flexible machine
x	492 492	205 199	492 492	65 260	199 707*

Table 3.11: At 147900, this mixture submits to the Flexible machine.

This situation goes on until cycle 177000 where a new machine learns to play fair with itself, and the Flexible machine hence reducing the dominating point of the Flexible machine. It does not play fair completely with itself hence the always-medium machine becomes stable. But this one persists unstably for very long time, waiting for the always-medium to happen.

0.99	x	always-low	always-medium	always-high	Flexible machine
x	494 494	356 199	491 496*	0 0	476 486

Table 3.12: At 177000, this machine keeps the Flexible machine at bay.

At cycle 225500, the always-medium actually happens. It plays fair with itself, the always-medium, always-low and Flexible machine. It gives little point to always-high. As you can see on the population average graph, this is the only other period in which the population payoff average reaches absolute level of fairness - where true 50-50 division holds, where everybody simply claims 5 from the first round onward, no sustained conflict (i.e. no war) with no probing, no retaliation, no waste of resources.

0.99	x	always-low	always-medium	always-high	Flexible machine
x	497 497*	497 199	497 497*	30 122	497 497*

Table 3.13: At 225500, the always-medium actually happens and it happens for some time.

At 270400, some bad luck just happens in the mutation hence the population payoff average reduces. At around 278400, a new mixture becomes stable, it does badly with everything else but good enough with itself. It gradually becomes fair with always-low and always-medium, not enough to destablise itself. It grows to be fair with Flexible machine also.

From 343100 to 350700, the machine is vulnerable toward a tough machine but a tough machine never emerges so it sustains its stability. In general, this one keeps the aggressive types at bay. It becomes less kind with always-medium also.

At 494800, it grows kind enough to always-medium for always-medium to take over but always-medium does not appear. It is vulnerable to the invasion of always-medium till 898000. This

0.99	x	always-low	always-medium	always-high	Flexible machine	tough 1	tough 2
x	470.5 470.5*	424.1 198.7	246.0 246.0	95.0 380.1	231.1 326.5	94.1 376.2	94.9 379.7

Table 3.14: At 898000, a complicated tough negotiator.

strategy does badly with everything else. It takes a prolonged period to probe and apologise and probe again to distinguish its own kind. The payoff is 470 because the sequence is 0 0 2 2 2 5 2 0 5 5 5 5 5 5... Eventually they settle down and claim 5 onward.

At 915000, there is a transition to another machine that is vulnerable to the invasion of always-medium. At 937500, the window for always-medium closes and another mutation opens a window

0.99	x	always-low	always-medium	always-high	Flexible machine	tough 1	tough 2
x	479.0 479.0	230.7 198.7	342.0 491.7*	96.9 387.6	294.8 258.6	96.9 387.6	100.8 400.2

Table 3.15: At 915000, a window opens for the invasion of always-medium.

for always-high to appear. If always-high appears it would definitely drops the population payoff average. However always-high does not appear till the end of the simulation.

0.99	x	always-low	always-medium	always-high	Flexible machine	tough 1	tough 2
x	479.0 479.0	263.9 198.7	342.0 491.7	192.8 771.1*	250.4 198.3	192.8 771.1*	100.8 400.2

Table 3.16: At 937500, a window opens for the invasion of always-high.

3.2 Characteristic evaluation and interpretation

At this point, I would like to develop a systematic characteristic test. I would try to identify the Flexible machine, the tough negotiator, the always-medium, and always-high. Also the one that on the face value is fair with itself and fair with always-medium hence it coexists peacefully in the population, however it is actually weak and unable to resist invasion. It submits to always-high. It is called a compromised strategy.

So the story goes that always-high never learns to share equitably with its own kind, hence one that learns to do so would meet the sufficient requirement to form an equilibrium. The necessity criteria is whether that one would be able to resist always-high and always-medium to keep the partnership stable. If it does not fulfil this second criteria, it is compromised, it invites always-high or always-medium into the population. If the compromised is popular enough, and always-high appears through the mutation process, always-high would take over the population. As always-high is in, the best response is always-low or the Flexible machine kind. This would form a mixture that can be stable but with very low population average. The always-low may eventually mutates a silent branch off the equilibrium path that be fair enough among itself. This leads to a path of the equitable share again if there is another mutation involves fighting off always-high (i.e. the aggressive kind).

There are two paths: one is the collapse of the perfectly equitable share. The direct collapse is made of the compromised kind that in the fair equilibrium does exactly what it is supposed to do hence no one can argue with it but it has the security breach off the equilibrium path that would not be able to resist the invasion of always high. However, the population may not fall directly into chaotic mix, it can be handled by a tough strategy which maintains a local maximum that is less than ideal but still far from chaos. This, however, requires another mutation that the strategy grow the ability to be aggressive. The other path to climb back from chaos to the simple way of life of sharing equitably is made of the Flexible machine kind (or always-low) who submits to always-high but has a silent branch reaching out to be kind to each other. It needs another mutation that is able to stand up against the aggressors. I suggest that the way back takes more mutations than the way to collapse directly and that path has a local maximum that stands firmly in the way: the tough negotiator that spends resources on limited war. This mechanism can be highly stable and it provides a reasonable way of channelling conflict which is waging limited wars. The conflicting period does not have to be initially, it can be initiated some time after the initial period. This mechanism sustains a better and more peaceful period than the chaotic one however it is still less ideal than the simple way of claiming 5 from the beginning, and as long as the game lasts.

One demonstration is from the simulation above: from randomness the population starts with an always-medium machine which is able to form an equilibrium with itself because it repels always-high and tough negotiator. This is what happens from cycle 0 to cycle 25000. Then this machine mutates in silent a branch that submits to always-high. We call the new machine a compromised one. It, on face value, acts fair with everyone. However it does not sustain a stable state for the population because it invites always-high. The thing is it is neutral to always-medium hence they coexist at any possible percentage of the population. The population can drift back and forth and once the percentage of this compromised machine is big enough, the population becomes vulnerable toward invasion of always-high. This happens from cycle 25000 to 29000. Once always-high is in, this compromised machine acts as an always-low. The mixture of aggressor and Flexible machine creates a period of chaotically low population payoff average. Because the match between always-high and the compromised one is fine, the match between the compromised one with itself is fine, but the match between always-high and always-high is zero. This is what happens at cycle 30000. This mixture is actually stable because it has the right mix of always-high. In cycle 31000, the mixture becomes unstable because there is not enough always-high in the population, it starts to be beneficial to be always-high. In cycle 32000, after a period of wasteful resources, the population develops a tough negotiator. The tough negotiator comes from the Flexible machine who submits to always-high but fair with itself. It has to grow so that it repels both always-medium and always-high. In other word, it has to become aggressive. It does not just keep the Flexible machines at bay, it exploits them. With 4 states, the strategy manages to distinguish all of the classic ones, and even some other tough machines. Impressive indeed.

An important note is that, during the extensive simulation, many times the machine is not stable, it is just waiting for mutation of a stable kind. Or it can happen that the population average payoff drops because of some bad luck mutation.

Simulation 1 b

I continue to run the simulation from the previous one. The number of cycle is reset to 0.

The situation at the end of the previous simulation is that there is a window for always-high to come in. It continues like that to cycle 23500 of this continued simulation where the strategy gives always-high a payoff of 771. The payoff it gives to always-high drops to 519 at cycle 25600. However, always-high is still the best response. The situation continues to cycle 77500 where the payoff it gives to always-high drops further down. This suddenly makes always-medium the best response because it gives always-medium payoff of 482.

0.99	x	always-low	always-medium	always-high	Flexible machine	tough 1	tough 2
x	479.0 479.0	219.4 198.7	291.0 481.9*	113.3 453.2	144.0 327.0	108.6 434.2	98.4 319.3

Table 3.17: At 77500, a window opens for the invasion of always-medium.

At cycle 128300, the population payoff average has a sudden drop, this makes always-medium a stronger best response.

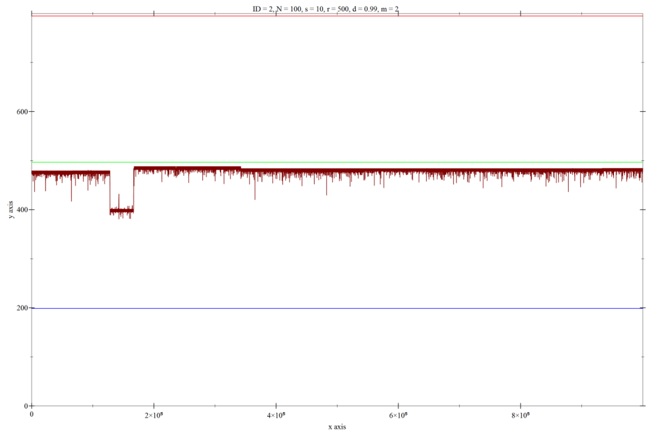

Figure 3.10: $\delta = 0.99$ the sequel

0.99	x	always-low	always-medium	always-high	Flexible machine	tough 1	tough 2
x	397.5 397.5	475.8 198.7	302.2 495.7*	63.8 255.1	189.8 316.3	69.4 263.5	88.6 336.7

Table 3.18: At 128300, a stronger invitation for always-medium.

At cycle 166900 the window for always-medium closes. The strategy regains its stability because it is less kind toward always-medium. It grows to exploit always-low and be fair with Flexible machine but remains stable until the end by doing badly with alway-medium and always-high. Many mutations of exploiting always-low come and go but they do not catalyse any change. Probably when one is already tough and aggressive, it does not change much the status quo by being more exploitive.

0.99	x	always-low	always-medium	always-high	Flexible machine	tough 1	tough 2
x	*438.9 438.9*	491.9 198.7	244.2 428.4	64.1 256.4	159.1 331.1	66.8 256.3	64.1 256.4

Table 3.19: From 166900 the window closes.

Simulation 2

I report here another run of the same settings, with brief study (Figure 7.11).

Figure a, at cycle 50000, the population achieves maximum payoff average, which means the payoff sequence is 5 everywhere. The machine at work is an always-medium. It has a lot of redundant states due to random mutation, but it starts with a state of playing Medium and whatever the opponent does, it remains in that state. So technically it is an always-medium.

At cycle 100000, the population continues to maintain the equitable sharing state, however, the mechanism to sustain this equitable state has mutated.

0.99	always-low	always-medium	always-high	Flexible machine	machine
always-low	199				
always-medium	497	497			
always-high	795	0	0		
Flexible machine					
machine	642	497	0	497	497

Table 3.20: benchmarking

From the table, the machine being matched with itself gives payoff 497 which means that it is absolutely equitable with itself. This is not enough to suggest that it has the character of an

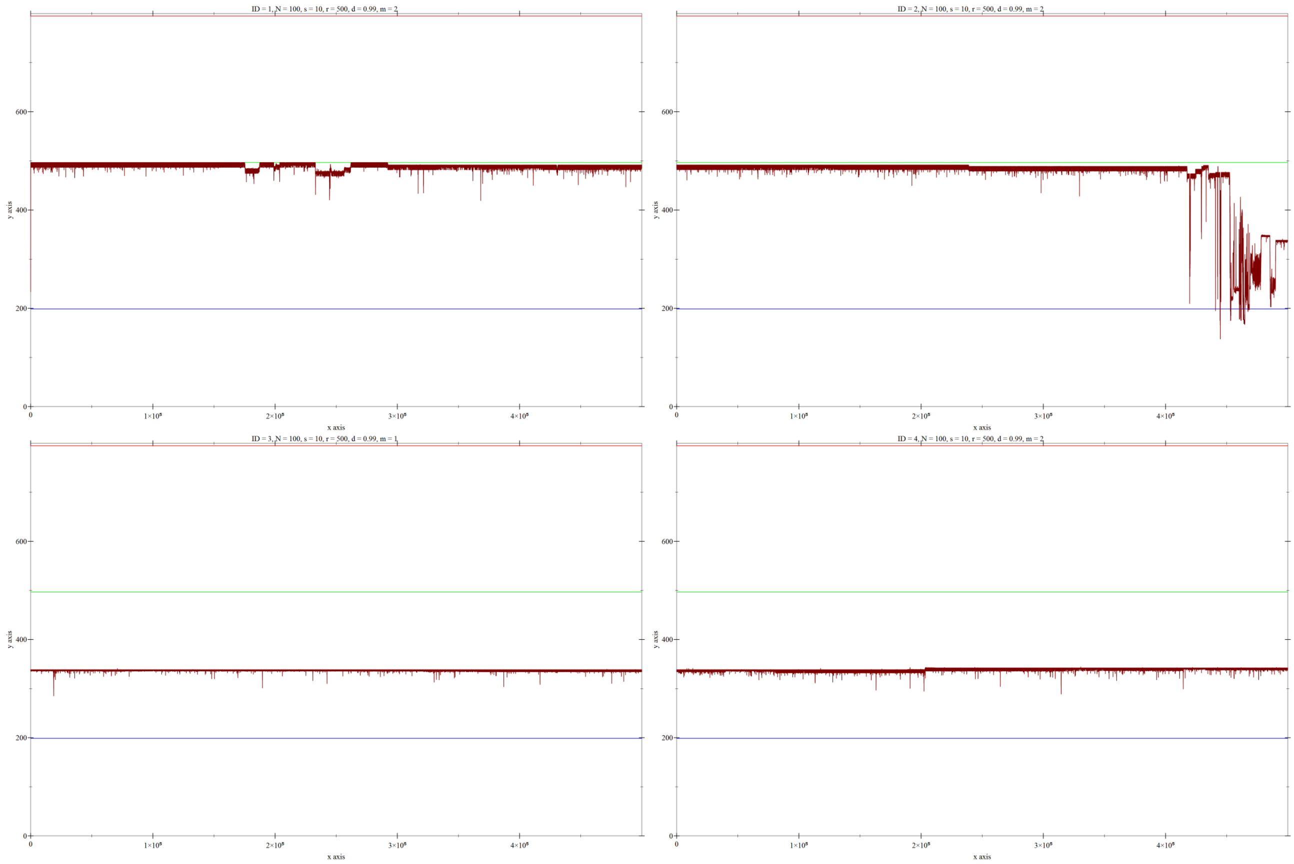

Figure 3.11: One typical run: $\delta = 0.99$.

always-medium. When being matched with the always-low, the always-medium gets the same payoff, but this machine gets 642. In fact, its payoff sequence is 5 5 5 8 5 8 5.. which means that this machine is able to recognise the exploitation opportunity toward the always-low, hence it jumps to claiming aggressively in every other round. Thus resulting in the payoff sequence of 5 5 5 8 5 8 5.. This machine never retreats from an always-high. It gets 0. But it plays fair and square with always-medium and Flexible machine. In general, this one has a reasonably accommodating scheme. It would play nice with fair types, learn to exploit a submissive type, but never retreat from an aggressive type. The last one is important because it is the criteria that makes it stable.

At cycle 150000, the machine becomes rigid again. It has many states but most of them play Medium, hence it acts like an always-medium. At cycle 240000, the population average is down a little bit. The payoff sequence that machines obtain is 2 2 0 0 2 5 5 5... Hence the initial rounds are spent for a series of probes before settling down to the equitable share. It plays Low for 2 rounds, then jumps to High for 2 rounds, then Low again, then fixes it at Medium.

0.99	always-low	always-medium	always-high	Flexible machine	machine
always-low	199				
always-medium	497	497			
always-high	795	0	0		
Flexible machine					
machine	493	493	199	488	478

Table 3.21: Benchmarking at 150000

This one is always an always-medium, except that it submits to always-high. The down period is not long though.

At 300000, the machine is an almost always medium, it plays the first round aggressive but immediately goes back to play fair.

0.99	always-low	always-medium	always-high	Flexible machine	machine
always-low	199				
always-medium	497	497			
always-high	795	0	0		
Flexible machine					
machine	505	488	0	486	491

Table 3.22: Benchmarking at 300000

At 400000, the machine does pretty well among itself except for the first round, however it does pretty bad against the benchmark set. It does not cooperate with always-medium, does not exploit always-low, does not submit to always-high, does not exploit or resist Flexible machine.

Simulation 3

I report here another run of same settings.

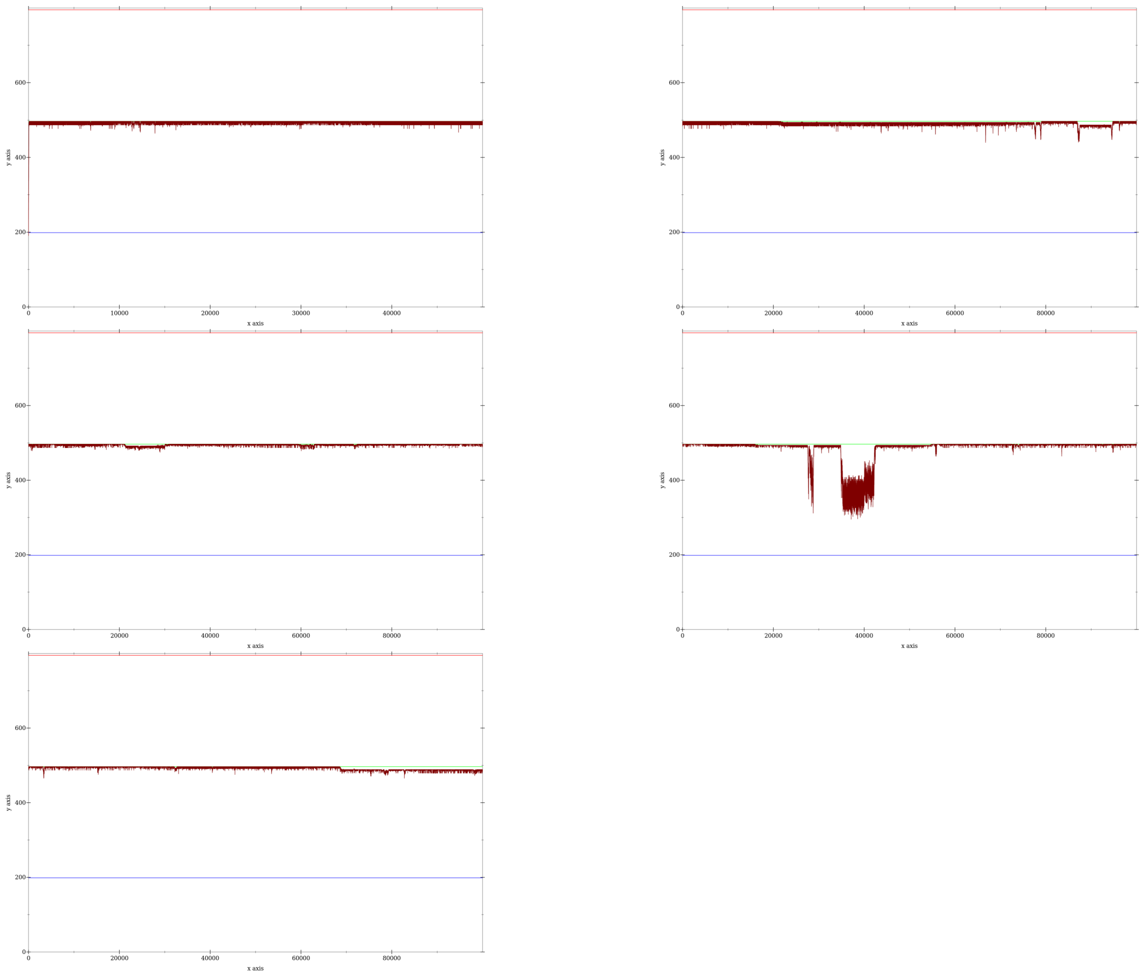

Figure 3.12: One typical run: $\delta = 0.99$.

3.2.1 $\delta = 0.9$

In this section I reduce the patience factor of the individual. The result changes numerically but not qualitatively.

Simulation 4

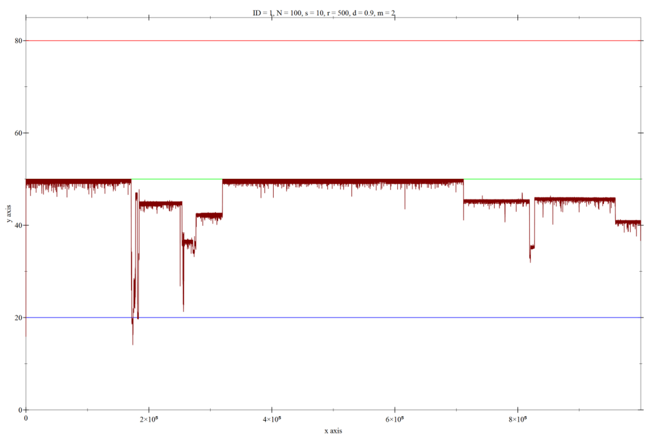

Figure 3.13: One typical run of 1 000 000 cycles: $\delta = 0.9$

From cycle 0 to cycle 9600, the strategy is an always-medium: it plays fair to itself, always-low, always-medium and Flexible machine. It gives always-high zero payoff. From cycle 10000,

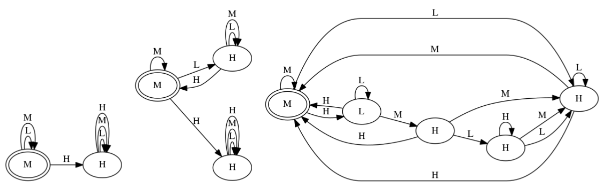

Figure 3.14: At 9600, 11100, 34200

there is a mutation that starts to exploit always-low. This mutation does not have any effect because it goes off equilibrium path. From cycle 33300, there is a mutation that starts to submit to always-high. And it exploits always-low less. In other words, it becomes less aggressive. This is dangerous because if it submits to always-high enough, always-high becomes its best response and the strategy loses its stability. However, at 71500, it suddenly exploits always-low again and gives always-high zero payoff. At 112500, it submits to always-high again and exploits always-low less. Finally, at 152800, it submits to always-high enough, and always-high becomes best response. This does not mean that the population average would drop immediately. This mutation makes the population vulnerable toward a hostile invasion. But it stays like that for a long time, waiting for the right mutation.

Eventually, at 171200 the population average starts to drop and it drops quickly. In 171500, the population average drops from 50 to 26.3, the reason is that the machine alternates between Medium and High with itself, hence the payoff sequence 5 0 5 0 5 0... However, this machine is not the best response to itself, the always-high is the best response to this machine because it submits to always-high completely. Some time after something similar to always-high appears.

In 172300, the population average is that of everybody playing low. The best response is alway-low. This mixture gives always-medium and always-high zero payoff. It is similar to an always-high strategy.

0.99	x	always-low	always-medium	always-high	Flexible machine	tough 1	tough 2
x	19.1 19.1	72.1 20.0	0.1 0.1	0 0	51.8 14.5	3.1 12.4	8.4 33.8*

Table 3.23: At 172300, the mixture is an always-high.

Notice from the benchmark table that the best response to this mixture would be the tough 2 machine. However, without a tough machine with complicated scheme, the simple always-low is a best response.

At 172400, the machine does badly among itself because it is too aggressive for its own good. The payoff sequence with itself is 0 0 0 5 0 5 0 5 0... This machine, however, would pair perfectly with an always-low machine.

At 174500, the population becomes less aggressive, it is more fair toward always-medium hence it appears to be vulnerable to the invasion of always-medium. But always-medium does not appear, the window closes. At 175500, it is vulnerable to always-high invasion. However, around 178400, the population mixture is able to get back to be fair with itself, always-medium and Flexible machine again. The population payoff average goes back to 47 which is almost perfect. They only waste the first round.

0.99	x	always-low	always-medium	always-high	Flexible machine	tough 1	tough 2
x	21.4 21.4	22.7 20.0	30.7 39.4*	2 8	23.7 26.7	2 5	7.6 30.4

Table 3.24: At 174500.

After some other bad luck mutation that drops the population payoff average further, at 184200 the population achieves something stable. If we look at the benchmark table, this one is a tough kind. It is fair enough to itself (except the first round), it exploits completely always-low and Flexible machine, resists completely always-medium and always-high. It resists the tough 1 machine however, not able to resist the tough 2. It is still a tough kind, though. This tough machine actually sustains a long period of stability for the society. No more chaos when conflict is channelled through limited wars. Around 220400, it has a mutation that reaches out to always-high and at 221000 it is no longer stable.

0.99	x	always-low	always-medium	always-high	Flexible machine	tough 1	tough 2
x	45.5 45.5	79.6 20.0	1/10 2/5	1.4 5.6	71.6 18.2	3.2 12.8	43.2 47.4*

Table 3.25: At 1784200.

0.99	x	always-low	always-medium	always-high	Flexible machine	tough 1	tough 2
x	45.0 45.0	*80.0 20.0	0 0	11.8 47.2*	*72.0 18.0	40.5 40.5	*45.0 45.0

Table 3.26: At 1784200.

At 228100, the machine mutates to start to be kind with always-medium. However it is still submitting to an aggressive kind (can be always-high but can also be a tough machine like the tough 1). The tough 1 machine never appears. Around 255100, the population payoff average drops further. Here is the payoff sequence between itself: 2 5 5 2 0 2 2 5 0 5 5 5... It is unstable toward always-high and sometimes always-medium or a tough kind but these kinds never appears. At some point the population payoff increases but it is still not stable.

Finally, at 319900, the always-medium is back. It has mutation that exploits always-low but this kind of mutation is not threatening. When one is strong and stable, a new feature of being exploitive probably would not destabilise its own crown. The dangerous mutation is the one that makes it less aggressive. After a very long time, at 685800, the machine grows a branch that starts to submit to always-high. At the same time, it exploits always-low less. This is a bad sign.

However, this time the collapse at 711900 is not directly to chaos, it is handled by the local maximum of the tough strategy.

0.99	x	always-low	always-medium	always-high	Flexible machine	tough 1	tough 2
x	*45.5 45.5*	46.5 20.0	27.2 33.6	7.9 31.8	38.0 33.5	*45.5 45.5*	40.5 40.5

Table 3.27: At 711900.

It is a neutral strategy with tough 1. This payoff is due to the sequence 5 0 5 5 5... It probes at the second round. At 819300 it collapses, it submits to Flexible machine:

0.99	x	always-low	always-medium	always-high	Flexible machine	tough 1	tough 2
x	33.2 33.2	45.0 20.0	30.8 32.8	6.4 25.5	29.3 47.5*	8.0 27.4	6.5 23.6

Table 3.28: At 819300.

The mechanism only becomes stable at 827400:

0.99	x	always-low	always-medium	always-high	Flexible machine	tough 1	tough 2
x	45.9 45.9*	48.8 20.0	43.8 45.9*	4.2 17.0	24.3 38.6	11.9 29.7	7.2 24.6

Table 3.29: At 827400.

This payoff is due to the sequence 5 5 0 5 5 5 5.... It probes at the third round. Overall, the tough strategy is able to probe at different round, not just the first one. And limited war is a local maximum we can rationally reach for in the time of crisis.

Simulation 4 b

I continue the simulation from the previous one and plot it here. Only the cycle is reset. We can safely assume the similar pattern with similar underlying story.

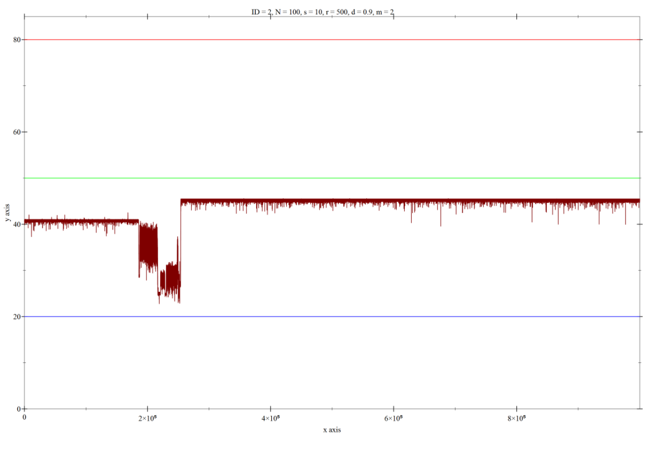

Figure 3.15: $\delta = 0.9$ the sequel

3.2.2 $\delta = 0.8$

When I reduce the patience level of the individual further, the frequency of the troublesome period reduces.

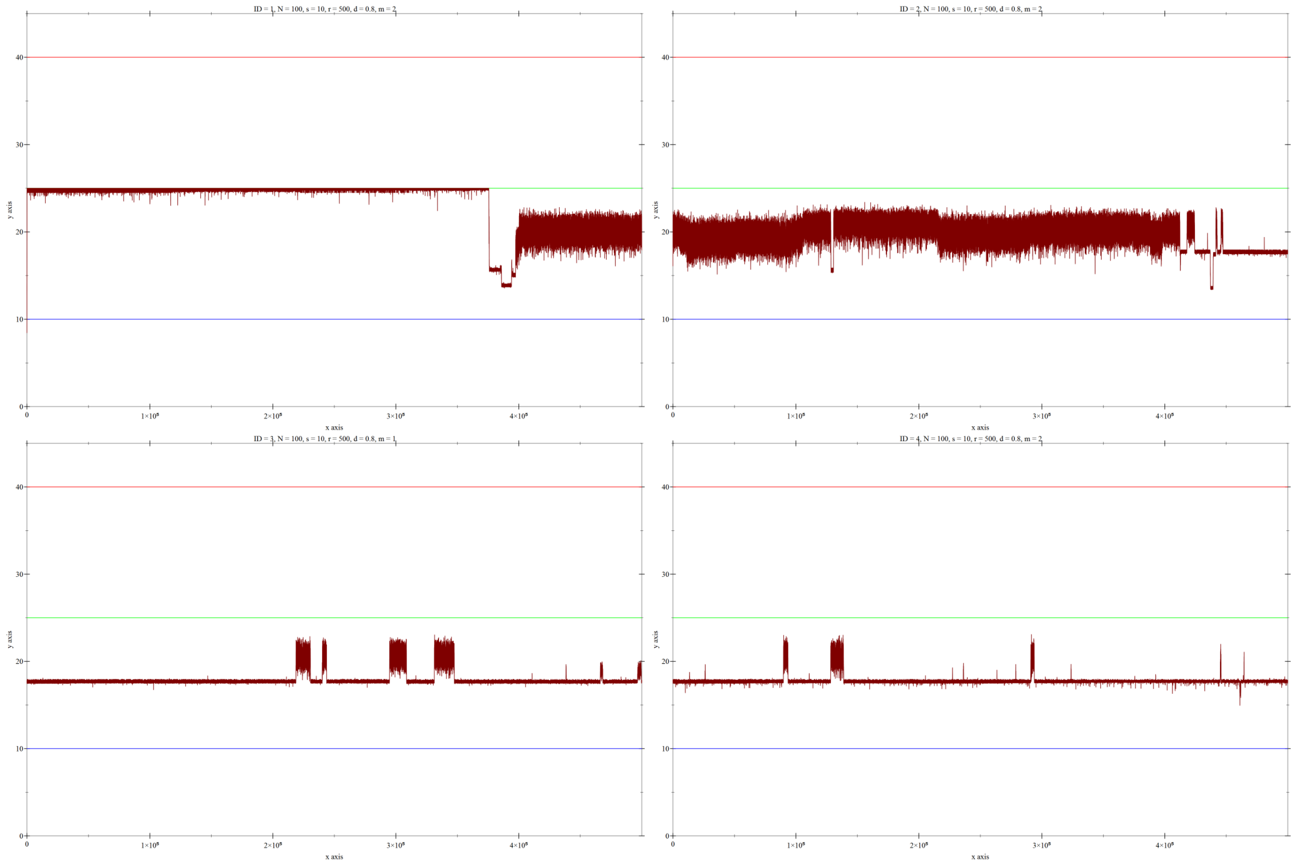

Figure 3.16: One typical run: $\delta = 0.8$.

Simulation 5

The violently fluctuating period that spans 400000 cycles (half the time) consists of a mixture of machines. The alternating payoff sequence of 0 2 8 5 5 8 2 .. suggests that they are somewhat flexible machines.

The flatly down period at the end features a single machine that claims High initially but eventually switching back to Medium. The payoff sequence is 0 0 0 5 5 5 0 5 5..

I suspect that there can be two stable states of non-equitable shares: a mixture of machines that alternate and a single machine that retreats after a while.

0.8	always-low	always-medium	always-high	Flexible machine
always-low	10			
always-medium	25	25		
always-high	40	0	0	
Flexible machine				

Table 3.30: benchmark

Simulation 6

Figure 7.17 shows another run with the same setting.

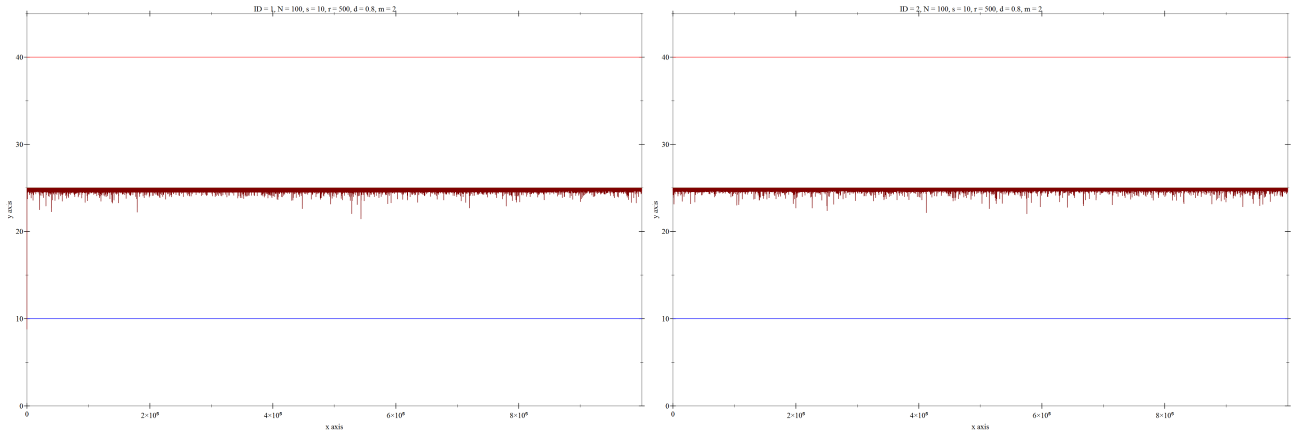

Figure 3.17: One typical run: $\delta = 0.8$. Total 2 000 000 cycles.

3.3 Myopic players

When the patience level of the individual homogeneously drops to 0.5, there is virtually no more ground for conflict. People just do not have time for that.

3.3.1 $\delta = 0.5$

Simulation 7

Figure 7.18 shows the result of 1000000 cycles, however, there is no sign of a collapsing of the 50-50 division norm.

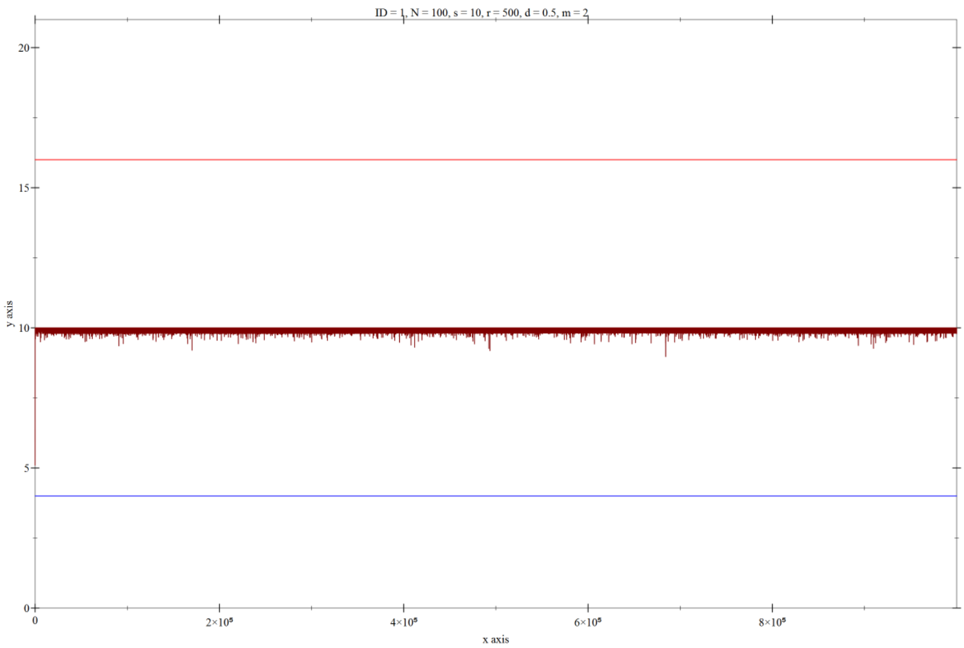

Figure 3.18: One typical run: $\delta = 0.5$.

3.4 Conclusion

For impatient players (i.e. as in the one shot Nash Demand game), the equitable share is highly stable. The mixture of claiming Low and High can theoretically stable, too because it is a mix equilibirum. However, in the replicator dynamics, the 50-50 division has so much larger basin of attraction.

When the virtue of patience increases, as shown in my simulation, it reduces the frequency of a perfectly equitable solution. However, it increases the appreciation for conflict in the form of limited war instead of going all for a chaotic mixture among unequals: the aggressor and the submitted party.

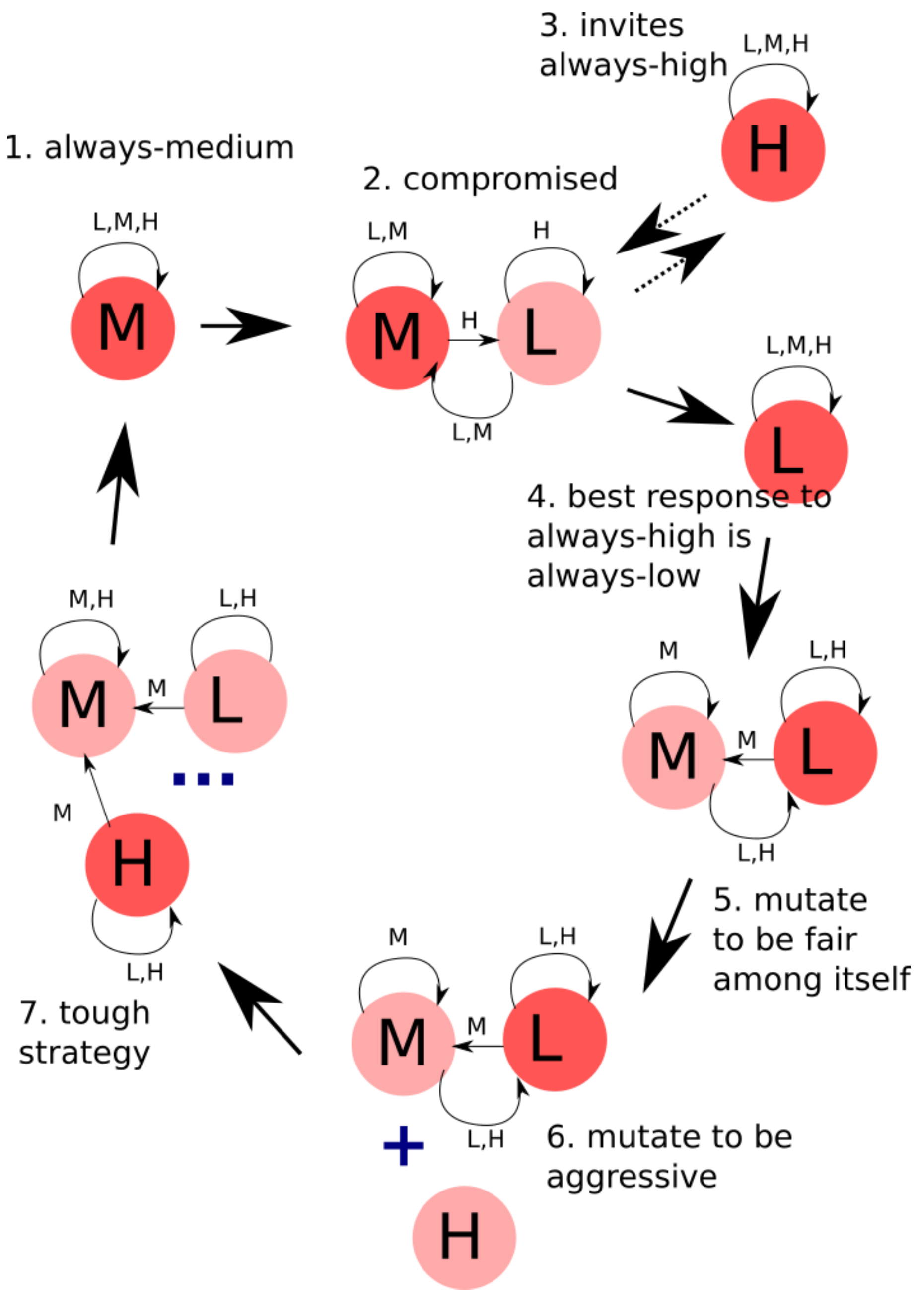

Figure 3.19: The mutation cycle of strategies: always-medium $->$ compromised $->$ mutates to be fair with its own kind $->$ learns to be aggressive $->$ tough strategy $->$ always-medium.

Chapter 4

The repeated Nash Demand game with a different approach: Markov chain in strategy structure

4.1 Strategy implementation

This game has 3 possible actions in each scene: to choose L, M or H. Hence we have an action scheme in the presentation of a vector (qL, qM, 1 - qL - qM). This vector contains the propensity to do each action. For example:

```
an action scheme (a vector): 0.5 0.5 0
```

This action scheme is to flip a coin on L and M, and 0 probability of choosing H.

We have 9 possible outcomes. The action plan for 9 possible outcomes can be arranged in a matrix-form table. In each cell, there is an action scheme. Hence the matrix would specify an action scheme after each possible scenario. Here is the structure of an action plan:

```
p1\p2 | L   | M   | H
--------------------------
L     | aLL | aLM | aLH
M     | aML | aMM | aMH
H     | aHL | aHM | aHM
```

aLL is the action scheme for the next round if this round's outcome turns out to be (L, L). For example:

```
p1\p2 | L       | M       | H
-------------------------------------
L     | (0 0 1) | (0 1 0) | (1 0 0)
M     | (0 0 1) | (0 1 0) | (1 0 0)
H     | (0 0 1) | (0 1 0) | (1 0 0)
```

According to this re-action plan, every time the opponent plays L, the strategy would prescribe to claim H in response. If the opponent plays M in this round, next round the automaton would also play M. If the opponent plays H in this round, the automaton retreats to L in the next round.

Another way to represent this structure:

```
initial action scheme: (action 0.5 0.5)
```

```
action plan:
(list
  (list (action 0 0) (action 0 1) (action 1 0))
  (list (action 0 0) (action 0 1) (action 1 0))
  (list (action 0 0) (action 0 1) (action 1 0))))
```

4.1.1 Classic automata

a. The one who always claim Low:

```
initial action scheme: (action 1 0)

action plan:
(list
  (list (action 1 0) (action 1 0) (action 1 0))
  (list (action 1 0) (action 1 0) (action 1 0))
  (list (action 1 0) (action 1 0) (action 1 0))))
```

b. The one who insists on claiming Medium:

```
initial action scheme: (action 0 1)

action plan:
(list
  (list (action 0 1) (action 0 1) (action 0 1))
  (list (action 0 1) (action 0 1) (action 0 1))
  (list (action 0 1) (action 0 1) (action 0 1))))
```

c. The one who always be aggressive (i.e. always claim High):

```
initial action scheme: (action 0 0)

action plan:
(list
  (list (action 0 0) (action 0 0) (action 0 0))
  (list (action 0 0) (action 0 0) (action 0 0))
  (list (action 0 0) (action 0 0) (action 0 0))))
```

d. The one who is flexible: if opponents claims L it claims H in the next round, if opponent claims M, it claims M in the next round, if opponent claims H, it takes the signal to claim L in the next round.

```
initial action scheme: (action 0 1)

action plan:
(list
  (list (action 0 0) (action 0 1) (action 1 0))
  (list (action 0 0) (action 0 1) (action 1 0))
  (list (action 0 0) (action 0 1) (action 1 0))))
```

4.2 Strategy interpretation

Take a random strategy R:

```
initial action scheme: (action 0.8 0.2)
action plan:
(list
  (list (action 0.3 0.6) (action 0.7 0.1) (action 0.7 0.1))
  (list (action 0.6 0.4) (action 0.0 0.0) (action 0.2 0.1))
  (list (action 0.1 0.1) (action 0.3 0.0) (action 0.2 0.4))))
```

At first sight I can derive some brief suggestion. For example, the initial action scheme is too conservative, with the propensity to claim Low is 0.8 and the 0 propensity to claim High. If the result is (L,L) then the scheme switches to being Modest, but if the result is (L,M) or (L,H) then the scheme is to stay shy with the propensity to claim Low is 0.7. It would not make further sense to comment too much on these numbers.

To make a better character evaluation of the overall scheme, I match this automaton with the classic automata. But first, I need to set some benchmark.

4.2.1 Benchmark

A sequence of all 2 for 500 rounds and $\delta = 0.99$ would returns the final payoff of 199. This is the first benchmark.

The second benchmark is the payoff sequence of all 5, resulting in the final payoff of 499. If an unknown automaton being matched with itself and gets this payoff, then naturally I take this as an indication of being always-medium character. There would be further questions before conclusion though. Because it can be of a Flexible machine kind who submits to Highs and exploits Lows.

The third benchmark is the payoff sequence of all 8, resulting in the final payoff of 795. If an unknown automaton being matched with Lows and returns this payoff, it is highly likely that it is of the same kind as Highs. Or else.

Let's take an example of evaluating the random automaton R above:
The first question in the character evaluation is:

- How kind (i.e. equitable) it is to its own kind?

The automaton would be matched with itself. The best case possible is the case of two always-medium playing against each other (resulting in payoff for both automata being 499).

Because the strategy R above is intrinsically stochastic, the payoffs would be different each time but in general, the result is 270. Which is 55 percent and is not very good.

As I elaborated in a previous chapter. The ability to form an equilibrium depends on the ability to resist the stable type always-medium or to resist the invasive always-high.
This leads to the next question:

- Does it resist always-high?

If yes, this path bears resemblance to a strategy that always claims High. I classify it as of the same aggressive kind.

To access the stability of the strategy, we can further ask:
- Does it resist always-medium?

These questions would be further illustrated.

4.2.2

There is another way to graphically interpret the strategy.

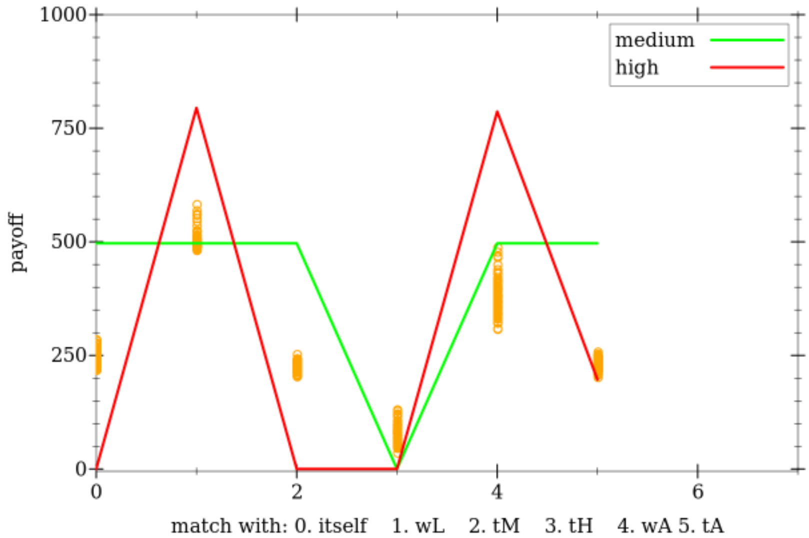

Figure 4.1: Automaton

On the graph, there are two lines: the green one belongs to the always-medium strategy, the red one belongs to the always-high strategy. These two classic strategies are important ones. The dots belong to the automaton of interest. Because its scheme is probabilistic, I run the matches many times to approximate its payoff. The y axis is the payoff.

On the x axis, position 0 is the payoff if that automaton matches with itself. Position 1 is how much the automaton can get from matching with an always-lows. This would show how exploitable the automaton is. The most exploitative kind is the always-high (which gets around 800). Position 2 is how much payoff the automaton gives to always-medium in a match with always-medium. If it is giving always-medium a lot of points, it loses its ability to become a stable state. However, if it gives always-medium very little point, it has a chance to be stable. The maximum one strategy can give to always-medium is to be perfectly equitable all the time, that is around 500. Position 3 is how much payoff the automaton would give to always-highs. If it submits to always-high and gives always-high a lot of points, it does not stand a chance to be stable on itself. But it can form a mixture with always-high that is stable and chaotic. In that case, it acts as an always-low. Position 4 is the payoff the automaton gets if it matches with a Flexible machine. Position 5 is the payoff it gives to the Flexible machine.

In the match with itself (position 0 on x axis), its performance is very close to the always-lows automaton but it is not that this automaton always claims Low. The final payoff can come from a sequence of 2 0 0 8 0 0 ... In the match with always-low, it does not exploit Lows but it acts like an always-medium. In the match with always-medium, it gives always-medium very little points. And it gives always-highs almost zero. In the match with the Flexible machine, its payoff is slightly higher than the flexible machine. Overall, this one is moderately aggressive.

Chapter 5

Simulation result: the repeated Nash Demand Game - Markov chain

5.1 Patient players

5.1.1 $\delta = 0.99$

Simulation 1

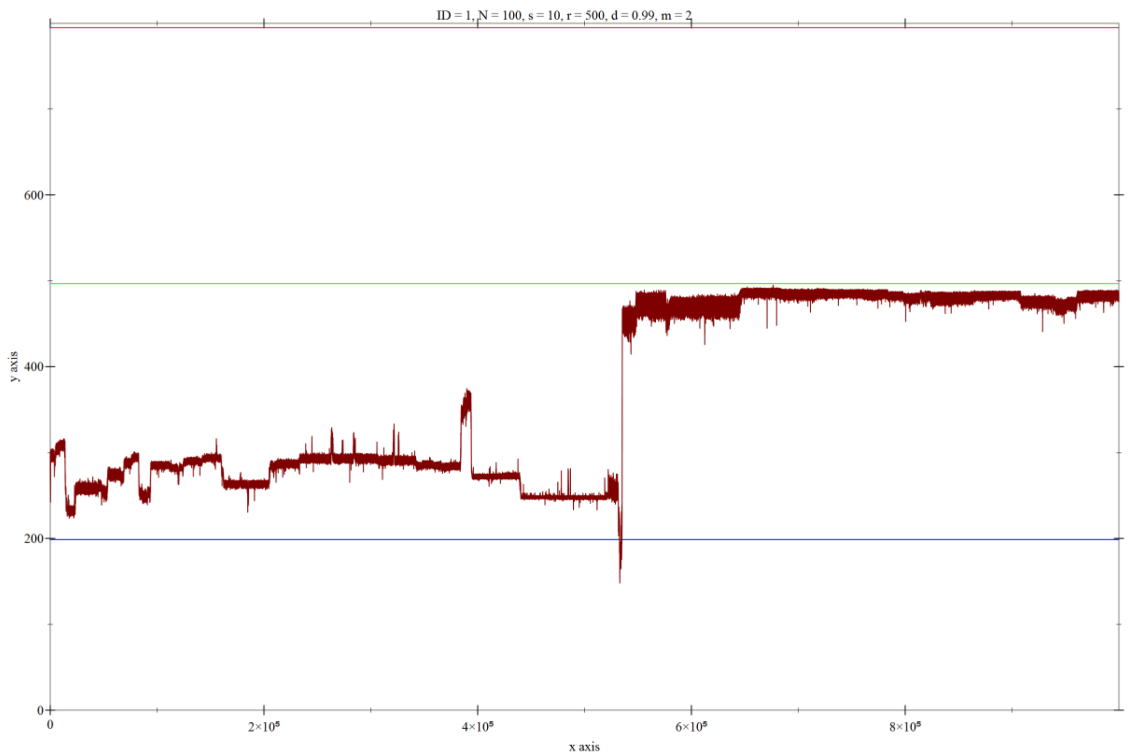

Figure 5.1: One typical run: $\delta = 0.99$, of 1000000 cycles.

From cycle 0 to cycle 83000, the population state is fluctuating but is not stable. This state can be easily invaded by an always-medium but always-medium never appears hence the window closes and the state manages to be stable at cycle 88200. Here is the portrait of the mixture:

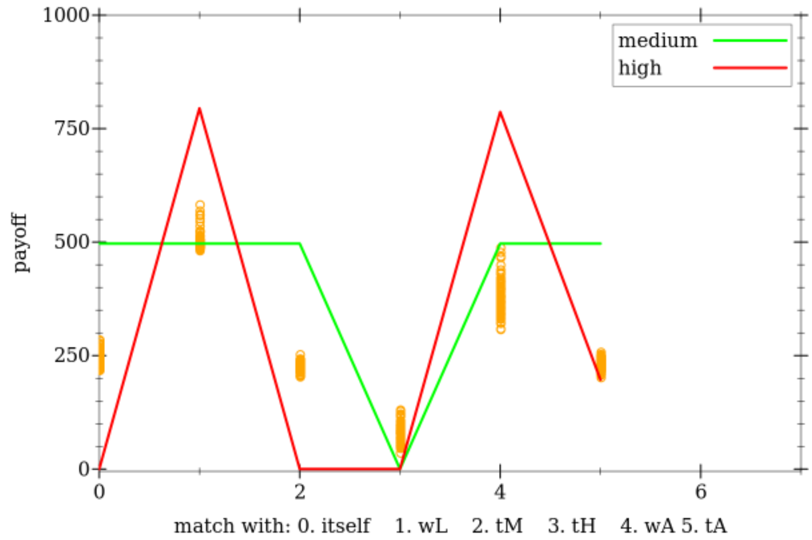

Figure 5.2: Automaton at cycle 88200.

First, the mixture gives itself payoff of around 250 (position 0 on axis x of the graph). It gives always-medium a little less than that (position 2 on axis x), but enough to make itself stable from the direct invasion of always-medium. It gives always-high zero hence is able to prevent always-high from entering the population. The point is, this one is not very kind to itself, it is only kind enough to maintain an equilibrium. The population payoff average would be similar to a chaotic situation (where the population is a mixture of always-high and always-lows). The payoff sequence with itself is 2 2 5 2 0 0 5 0 5 0 8...

At 111000, it gives itself more points, exploits always-lows more and gives always-high even less payoff. The payoff sequence with itself is 8 0 5 8 8 0 2 0 5 2 2 ...

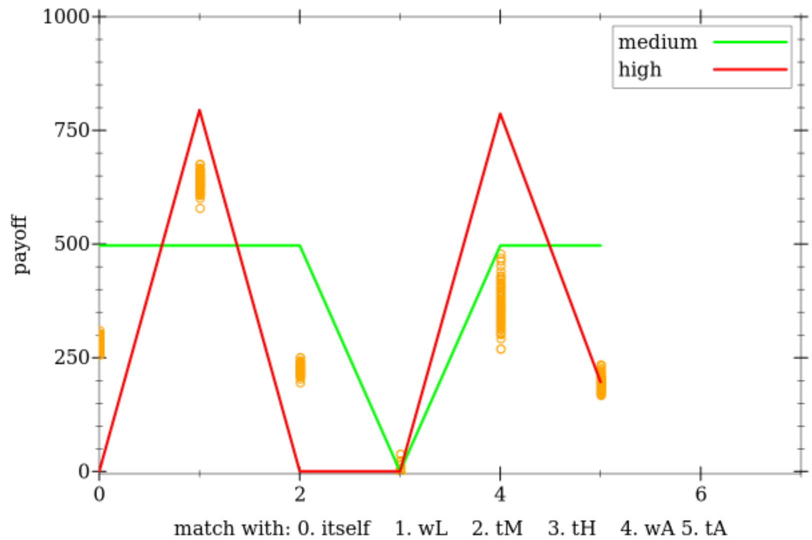

Figure 5.3: Automaton at cycle 111000. It is aggressive.

```
(initial action 0.9 0.0)
 (list
```

```
  (list (action 0.3 0.7) (action 0.1 0.0) (action 0.3 0.5))
  (list (action 0.0 0.3) (action 0.2 0.2) (action 0.0 0.8))
  (list (action 0.2 0.0) (action 0.3 0.7) (action 0.0 1.0))))
```

This automaton starts out playing Low. If the opponent plays Low, its action scheme is quite aggressive: see the first column with aLL = 0.3 0.7, aML = 0.0 0.3 and aHL = 0.2 0.0 which means that after (L,L) it plays Medium with probability 0.7, after (M,L) it plays High with probability 0.7 and after (H,L) it plays High with probability 0.8.

If the opponent plays Medium (the second column), this one is also aggressive. After LM, the probability to play High is 0.9. After MM the probability to play High is 0.6. After HM the probability to play Medium is 0.7.

If the opponent plays High (the third column), this one has a propensity to claim Medium. After LH, it plays Medium with probability 0.5. After MH it plays Medium with probability 0.8. After HH it plays Medium with probability 1.

Overall this one starts out Low but its behavior ranges from fair to aggressive.

The situation does not change significantly until 334000 where the population state loses its stability because it starts to give the Flexible machine a lot of points. At 348500, it gives always-high a lot of points. It is no longer stable.

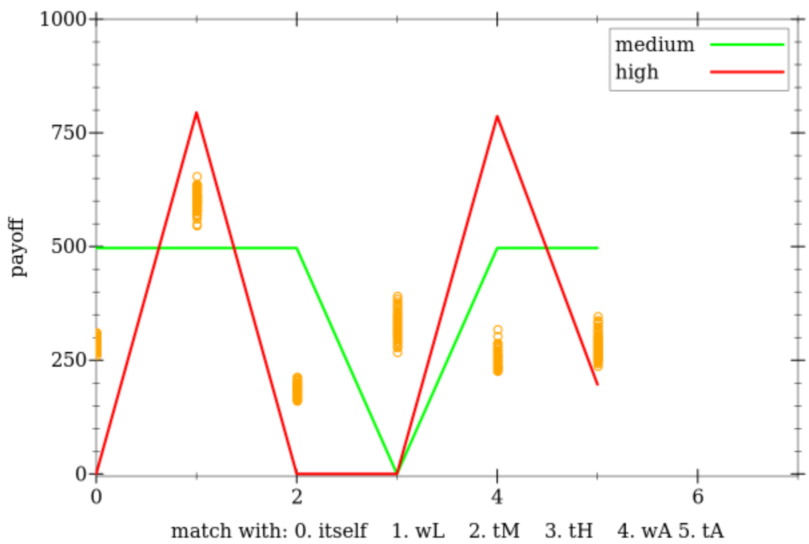

Figure 5.4: Automaton at cycle 348700.

```
(action 0.9 0.0)
 (list
  (list (action 0.3 0.1) (action 0.2 0.0) (action 0.5 0.4))
  (list (action 0.9 0.1) (action 0.1 0.0) (action 0.7 0.0))
  (list (action 0.1 0.3) (action 0.2 0.5) (action 0.0 1.0))))
```

If we look into its scheme, we see that, after MM, it plays High with probability 0.9 and after HH it plays Medium with probability 1. This one alternates between Medium and High with itself hence the payoff sequence it gives itself would settle to be 0 5 0 5 0 5... This reduces its payoff a lot. And it is giving always-high a lot of 8. This one invites always-highs into the population.

At 394700, the automaton is able to resist always-high and form a stable equilibrium with itself.

```
(action 0.0 0.0)
```

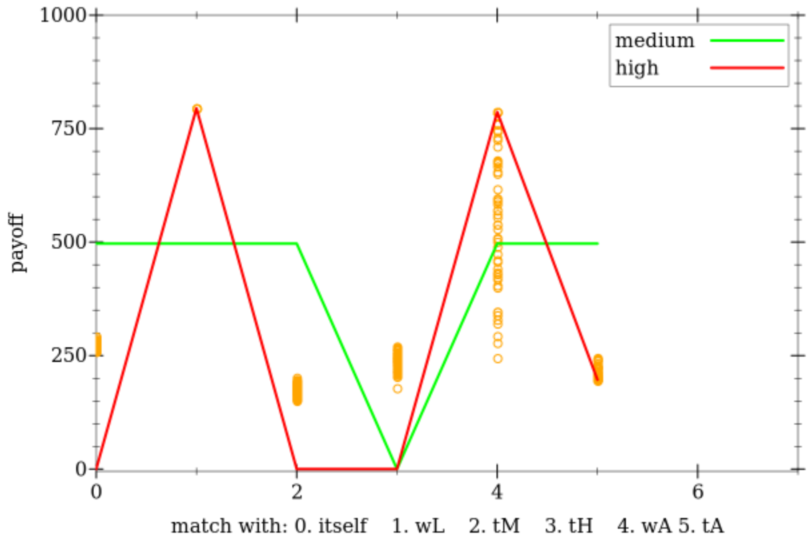

Figure 5.5: Automaton at cycle 394700.

```
 (list
  (list (action 0.3 0.1) (action 0.1 0.0) (action 0.2 0.1))
  (list (action 0.9 0.1) (action 0.1 0.0) (action 0.7 0.0))
  (list (action 0.0 0.0) (action 0.3 0.5) (action 0.0 1.0))))
```

It starts out to play High absolutely. After a LH, it continues to exploit with certainty. After a HH, it plays Medium with probability 1. This one is in general aggressive, except two cases. After a ML, it plays Low with probability 0.9, after a MH it plays Low with probability 0.7.

The payoff sequence it gives itself is an alternate sequence of 0 and 5. It explains why the population average payoff is so low.

From 440300, the population state fluctuate between stable and unstable because the relation with always-high: sometimes it gives always-high more than it gives to itself, sometimes less.

One thing to note is that, of all these time, the automaton is very exploitive to always-low. However this trait does not seem to have any effect.

Around 535500, the situation changes drastically, the population average payoff suddenly jumps from 200 to 500. This is a swing from the lowest to the highest.

```
(action 0.1 0.0)
 (list
  (list (action 0.2 0.3) (action 0.0 0.0) (action 0.8 0.0))
  (list (action 0.0 0.1) (action 0.0 1.0) (action 0.3 0.1))
  (list (action 0.0 0.0) (action 0.0 0.1) (action 0.0 0.4))))
```

This is a remarkable automaton. It starts out to play High with probability 0.9. After a HL, it continues to play High (i.e. be exploitive) with certainty. Hence this one is able to exploit always-low and flexible machine completely.

After a LH, it continues to play Low with probability 0.8. After a HH, it does not retreat at all. Hence it gives always-high insufficient points.

After a MM, it plays Medium absolutely. With itself, the payoff sequence is 0 0 0 0 0 5 5 5 5 5 5... Medium is an absorbing state, hence after a while, it figures out and plays Medium forever. This one is a tough machine. Only that it is probabilistic so sometimes it figures out with always-medium earlier than it does with itself. Hence this one is not completely able to resist always-medium.

Then evolution finds a remarkable way to resist always-medium.

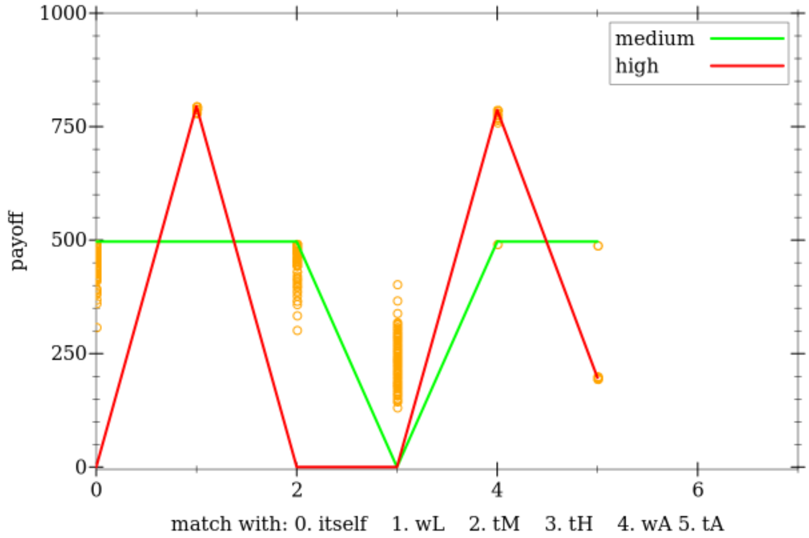

Figure 5.6: Automaton at cycle 535500.

iNDG	Automaton	Always-low	Always-medium	Always-high	Flexible
Automaton	*468.5 457.2*	*794.7 198.7	409.8 409.8	46.4 185.7	*786.7 196.7
always-low	198.7 794.7*	198.7 198.7	198.7 496.7	*198.7 794.7*	198.7 791.7
always-medium	453.5 453.5	496.7 198.7	*496.7 496.7*	0 0	496.7 496.7*
always-high	160.8 40.2	*794.7 198.7*	0 0	0 0	*786.7 196.7
flexible	196.7 786.7*	791.7 198.7	*496.7 496.7	196.7 786.7*	496.7 496.7

Table 5.1: Almost tough automaton at 535500.

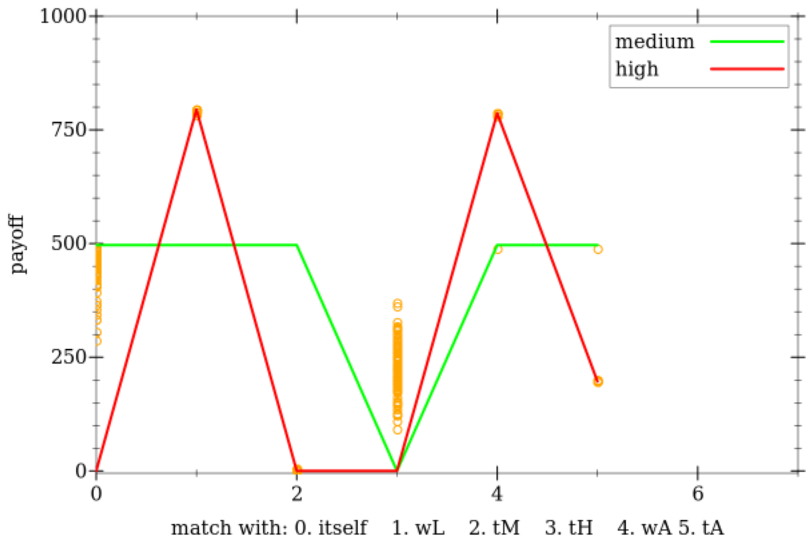

Figure 5.7: Automaton at cycle 541900.

```
(action 0.1 0.0)
 (list
  (list (action 0.2 0.3) (action 0.0 0.0) (action 0.8 0.0))
  (list (action 0.0 0.1) (action 0.0 1.0) (action 0.3 0.1))
  (list (action 0.0 0.0) (action 0.0 0.0) (action 0.0 0.4))))
```

It figures out a way with itself, the payoff sequence is 8 8 8 8 8 0 0 5 5 5 5.... It happens like this: it starts out playing High at 0.9 and after a HL, it plays High with certainty. However, after

iNDG	ToughAutomaton	Always-low	Always-medium	Always-high	Flexible
Tough Automaton	*448.9 448.9*	*794.7 198.7	0 0	28.2 112.9	*786.7 196.7
always-low	198.7 794.7*	198.7 198.7	198.7 496.7	*198.7 794.7*	198.7 791.7
always-medium	0 0	496.7 198.7	*496.7 496.7*	0 0	496.7 496.7*
always-high	168.7 42.2	*794.7 198.7*	0 0	0 0	*786.7 196.7
flexible	196.7 786.7*	791.7 198.7	*496.7 496.7	196.7 786.7*	496.7 496.7

Table 5.2: Tough strategy at 541900. This is the one that stabilises the almost-equitable norm after 500000 cycles.

a LH, it plays Low with probability 0.8. Hence after a few rounds of 8s, it starts to get 0 because of two Highs. After a HH, it plays Medium with probability 0.4 and after a MM, it sticks to Medium with absolute determination.

This one never gives points to always-medium though. Because after a HM, the probability to play H is 1. Hence the match with always-medium is a forever sequence of HM that no one is going to find any way to get out of the loop.

This one is a tough strategy. It finds ways to distinguish always-high, always-medium to resist them, and it recognises itself to be fair. It exploits always-low and flexible completely. All these courses of actions using the same recipe (!). Incredibly clever.

After that, for a period of time, the state fluctuates between stable and unstable because the strategy fluctuates between giving always-medium nothing and giving always-medium more points than it should do. This destabilising mutation happens at the initial action scheme. The automaton starts out playing Medium at probability 0.8. And once it is absorbed into the state of both playing M: (M,M) it stays there with absolute commitment. The mutations in this kind of model happens faster than the one using the finite state machine.

Anyway, here we are witnessing a special structure inside an automaton: an absorbing state of (M,M) and an action scheme of playing High absolutely after the outcome of (H,M) or after (H,H). This is the mechanism to distinguish between itself and an always-medium. If initially it plays High and the other keeps playing Medium rigidly, it would continue to play High rigidly. This never ends the payoff sequence of zeros for both sides hence it is able to repel always-medium completely. But with itself, after a few rounds of probing, it would be able to be absorbed into the state of playing (M,M) forever.

There is another thing that matters about this plan: the initial action scheme. For example, with an initial action scheme that is aggressive: (0.1 0.2) (i.e. playing High with probability 0.7 in the first rounnd) the plan works very well. On the other hand, with a fairly square initial action scheme, say, (0 0.5) (i.e. in the first round, toss a coin between Medium and High): if the automaton plays Medium in the first round, it would be immediately absorbed into the state of (M,M) with always-medium forever and it would gives always-medium maximum points of the fully equitable share. This would obviously destabilise itself because with itself, it spends some time probing and retaliating.

The situation remains the same until the end of the simulation.

Simulation 1 b

I suspect that the situation is the same: one tough automaton that has the contingency to distinguish between itself and always-medium. It seems that the compromised strategy never makes it in this kind of system. Because of two factors: first, the initial scheme is Low and second the reaction scheme after a (L,H) is also submissive. However, reaction scheme toward a hostile move is hostile in general so when the initial scheme is low, the automaton always comes back

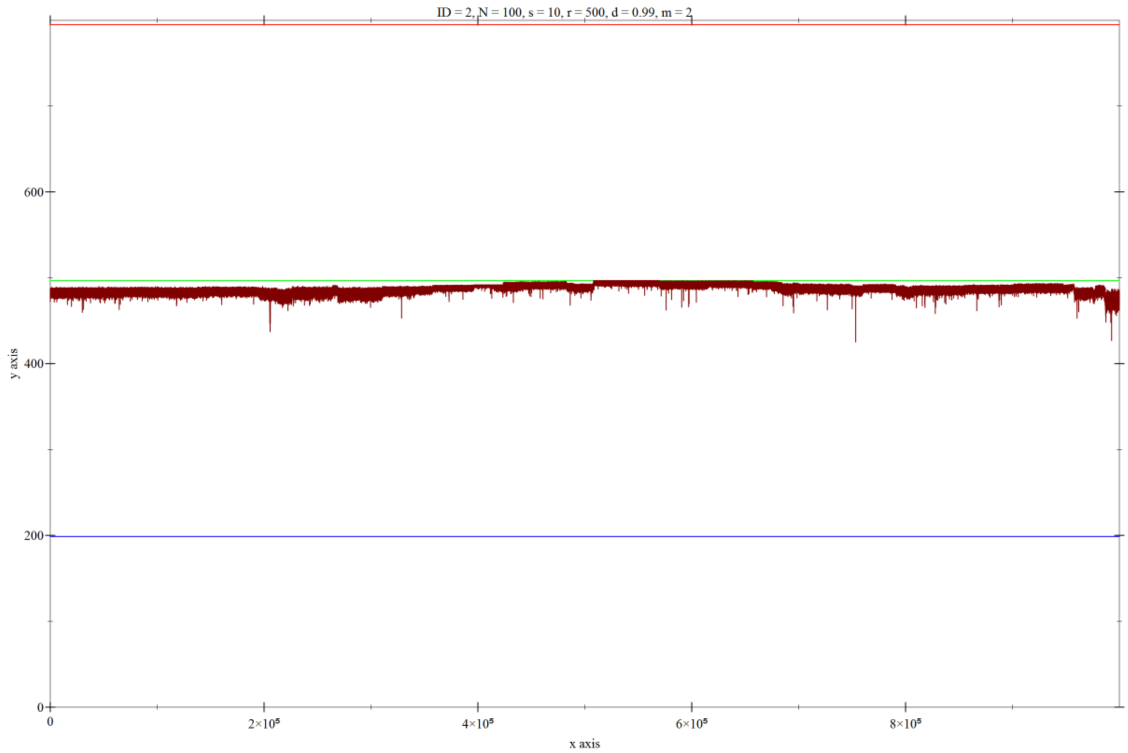

Figure 5.8: Simulation 1 continued: $\delta = 0.99$.

aggressively. And when the reaction scheme after a (L,H) is low, it does not have a way to enter that state, and even if it does, the reaction scheme is probabilistic, hence it would find a way out.

The stabilising of the chaotic mixture (between aggressor and submissive one) is harder to achieve in this stochastic setting. It is so much harder to pinpoint down the compromised strategy long enough to collapse the equitable norm. The local maximum of a tough automaton, hence becomes harder to escape. It is easier to reach and harder to escape. This is why we almost never see the fully equitable share norm in the simulation. We also never see a fully chaotic collapse.

Simulation 2

When the population reaches the level of payoff that is equal to the fully equitable norm, in a rare case, it could be due to the fact that the automaton alternate between L and H. The action scheme after a (L,H) would be to play H absolutely (0 0) and the action scheme after a (H,L) would be to play L absolutely (1 0). This could be reached by mutation. However the initial scheme is also very important to be able to lead the automaton down that coordinated road. The initial plan cannot be certain because that would make it impossible to coordinate because the strategy would be matched with itself. In that case, it goes High and Low together and creates disaster.

In Figure a, from very early the situation is already stable. The population payoff average is very low, though because the automaton seems to alternate resulting in the payoff sequence of 2 0 0 5 8 8 2 2 8 0 0... For example, at cycle 1400, here is the automaton:

```
(action 0.2 0.1)
 (list
  (list (action 0.2 0.0) (action 0.1 0.1) (action 0.4 0.0))
  (list (action 0.5 0.3) (action 0.8 0.1) (action 0.4 0.1))
  (list (action 0.7 0.1) (action 0.7 0.2) (action 0.1 0.8))))
```

This one starts out to be aggressive (the probability to play High is 0.7). After a (H,H) the

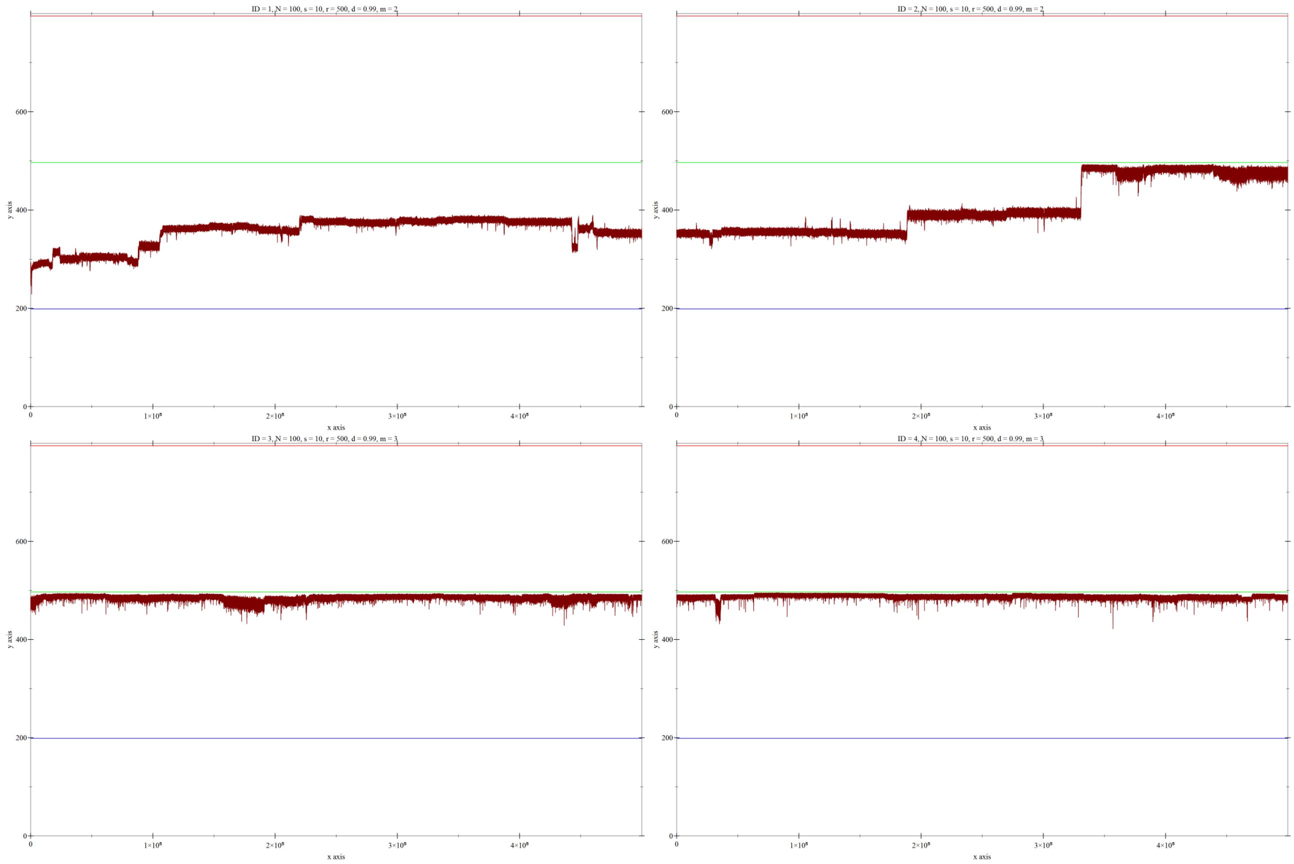

Figure 5.9: One typical run: $\delta = 0.99$

probability to play Medium is 0.8. After a (M,M) the probability to play Low is 0.8. After a (L,L) the probability to go High together is 0.8. Because of the probabilistic nature, sometimes the outcome is (L,H), after a LH, the action scheme is (0.4 0) which means that they probably go to (H,L). Hence the resulting payoff sequence of alternating payoffs. This action plan is stable though because it gives less points to other automata than it gives to itself.

After a while, the alternating scheme refines to be so much better. They coordinate smoother and there are less 0s in their payoff sequence. Hence the population payoff average raises. For example, at 21300:

```
(action 0.2 0.1)
 (list
  (list (action 0.2 0.0) (action 0.1 0.1) (action 0.0 0.0))
  (list (action 0.5 0.3) (action 0.7 0.1) (action 0.4 0.1))
  (list (action 0.7 0.1) (action 0.7 0.2) (action 0.1 0.8))))
```

This one starts to play High with probability 0.7. After a HH, it plays Medium with probability 0.8. After a MM it plays Low with probability 0.7. After a LL it plays High with probability 0.8. The probability after LH has changes to (0 0) which means that after a LH it plays High with probability 1. After an HL, it plays Low with probability 0.7. This action plan results in a payoff sequence of 0 5 2 0 5 8 0 0...8 2 8 2 8 2...

The action plan is refined further and the payoff sequence is smoother. For example, at 107000,

```
(action 0.0 0.0)
 (list
  (list (action 0.4 0.0) (action 0.1 0.5) (action 0.0 0.0))
```

```
  (list (action 0.5 0.4) (action 0.2 0.6) (action 0.4 0.0))
  (list (action 0.8 0.2) (action 0.3 0.3) (action 0.0 1.0))))
```

This one starts out to play High with certainty. After HH it plays Medium with certainty. After MM it plays Medium with probability 0.6. If it manages to enter a state of alternating between H and L, it seems to stay there. Because after a LH it plays High with certainty and after a HL it plays Low with probability 0.8. This is a mixture of High and Low inside a single scheme. This interesting feature is possible because of the probabilistic nature of the strategy. The same ex ante action plan can land on different actions ex post. And evolution cultivates an action plan that has a path leading to alternating sequence of 8 and 2. The initial phase before the alternating is required though, and it is not efficient (though it has become better) resulting in the payoff sequence of 0 5 0 5 0 8 2 0 8 2 8 2 ... This is a remarkable feature that evolved out of chaos in my opinion.

Another tweak to this could be that it starts out with the action scheme of (0.8 0) which means that it either play Low with probability 0.8 or play High with probability 0.2. After a LL, it flips a coin to either play Low or High (the action scheme is 0.4 0). This does not introduce Medium into the sequence. After a LH it plays High with probability 1. After a HL it plays Low with probability 0.8. After a HH, it plays Medium with probability 1 and after MM it has the action scheme of (0.2 0.6). This makes the payoff sequence better and if lucky the resulting sequence is 2 2 8 2 8 2 8 2...0... with the 0 introduced so much later into the game. This, of course, matter because the later the 0 the better.

```
(action 0.8 0.0)
 (list
  (list (action 0.4 0.0) (action 0.3 0.5) (action 0.0 0.0))
  (list (action 0.5 0.4) (action 0.2 0.6) (action 0.8 0.0))
  (list (action 0.8 0.2) (action 0.3 0.2) (action 0.0 1.0))))
```

At 310300, the initial action scheme is to play Low with probability 1:

```
(action 0.1 0.0)
 (list
  (list (action 0.3 0.5) (action 0.3 0.4) (action 0.0 0.0))
  (list (action 0.2 0.6) (action 0.4 0.2) (action 1.0 0.0))
  (list (action 0.8 0.2) (action 0.0 1.0) (action 0.0 1.0))))
```

After a LL, the probability to play Medium is 0.5. After a MM, the probability to continue to play Medium is only 0.2 but the probability to play High or Low is equally 0.4. This action scheme encourages a start of alternating sequence between 8 and 2. And then, to help the matter, after a LH, the probability to play H is 1. After a HL, the probability to play Low is 0.8. After a MH and both getting zero, the plan is to retreat to play Low with certainty. After an HM, the plan is to retreat to play Medium with certainty. Hence after an outcome of MH the two automata (executing the same action plan) can coordinate an LM smoothly giving both points rather than 0.

In Figure b, I continue the simulation but resetting the cycle number back to 0. At cycle 100000, the automata engage in a long sequence of 0 5 0 5... (alternating between H and M) before settling down into the pattern of alternating between H and L. The payoff sequence is 0 5 0 5 0 5...8 2 8 2...

```
(action 0.0 0.8)
 (list
  (list (action 0.2 0.4) (action 0.5 0.3) (action 0.0 0.0))
  (list (action 0.2 0.0) (action 0.2 0.0) (action 0.1 0.0))
  (list (action 0.8 0.2) (action 0.0 0.1) (action 0.0 1.0))))
```

At 331800:

```
(action 0.5 0.4)
 (list
  (list (action 0.0 0.3) (action 0.2 0.3) (action 0.0 0.0))
  (list (action 0.0 0.9) (action 0.2 0.0) (action 0.1 0.3))
  (list (action 1.0 0.0) (action 0.0 0.8) (action 0.0 1.0))))
```

After a LH, the probability to play High is 1 and after an HL, the probability to play L is 1. With this action plan, if the automata manages to get into the flow, it stays in the alternating sequence of 8 and 2 absolutely. Hence you can see the sudden raise to almost the best possible population payoff average. This population payoff average is similar to the one of perfectly equitable share, however the mechanism is completely different. It is a sequence of alternating between 8 and 2 instead of a sequence of 5s. Other action schemes in this strategy also helps to enter the zone easier: after a LL, the probability to play High is 0.7, after a HH, the probability to play Medium is 1, after a MM the probability to play High is 0.8 which leaves room for one automaton to take the path of playing High and the other to play Low. Once they are absorbed into the sequence, it is inescapable.

I suspect that the situation stays like this till the end of the simulation (including figure c and d). This one is very hard to beat and it is very hard to lead to the collapse of a chaotic mixture. It is a mixture but it is actually a well coordinated mixture instead of creating chaos.

Simulation 3

I report in Figure 9.10 another simulation run with the same setting.

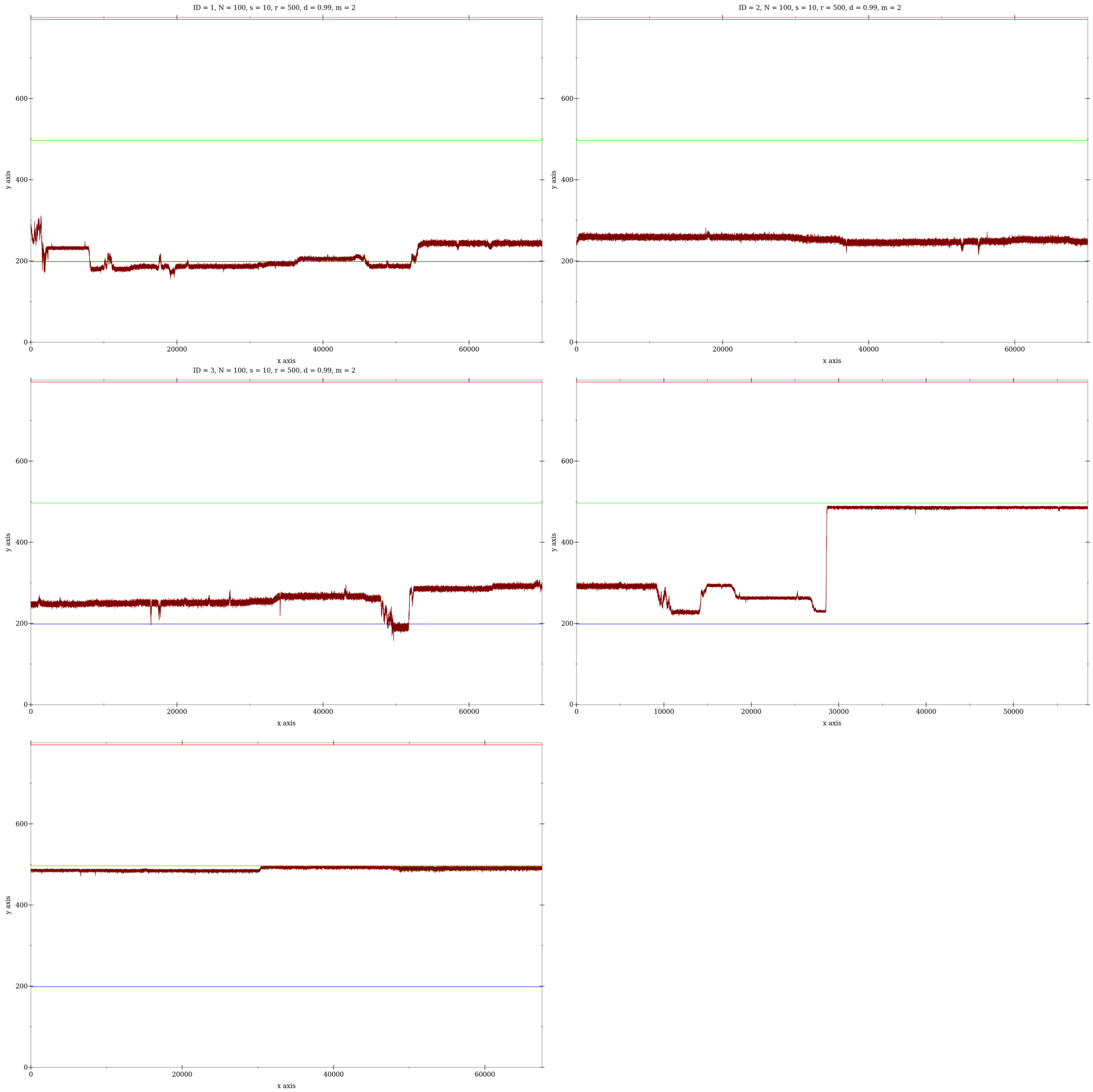

Figure 5.10: One typical run: $\delta = 0.99$.

5.1.2 $\delta = 0.95$

Simulation 4

Figure 9.11 a, at 205400, the strategy stabilises with similar mechanism: they alternate between 8 and 2 but with a lot of 0s in the payoff sequence. At 401700 they coordinate smoother hence the population payoff average raises. However, the strategy at this point is giving always-medium a lot of points. Hence it is on the edge of being destabilised. At 601700, the compromised scheme toward always-medium ceases. The automaton has evolved the action scheme that would generate the definitive alternating sequence of 8 and 2:

```
(action 0.4 0.4)
 (list
```

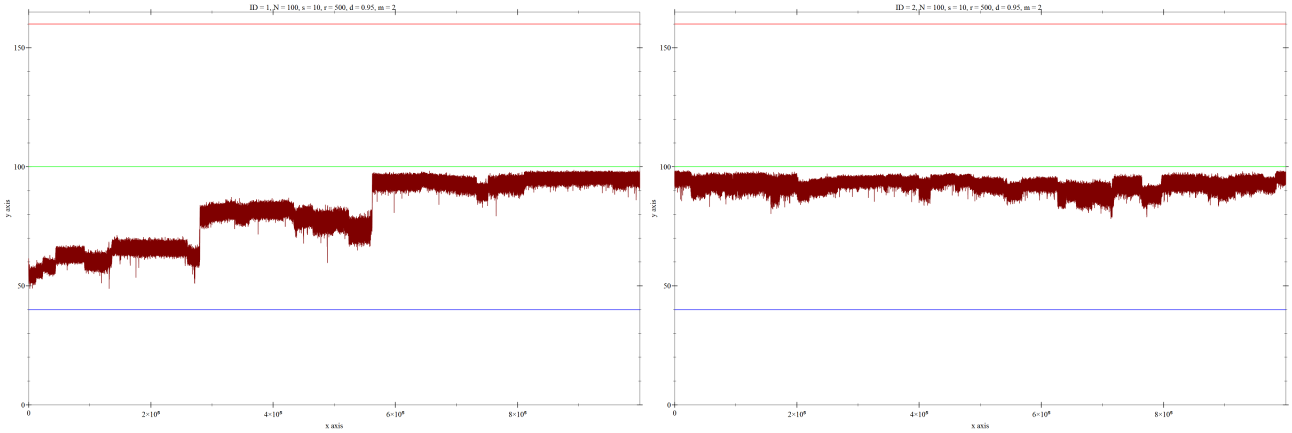

Figure 5.11: One typical run: $\delta = 0.95$

```
(list (action 0.1 0.3) (action 0.1 0.9) (action 0.0 0.0))
(list (action 0.4 0.3) (action 0.2 0.0) (action 0.9 0.1))
(list (action 1.0 0.0) (action 0.6 0.0) (action 0.3 0.1))))
```

We can see that after a LH, the probability to play H is 1 and after HL the probability to play Low is 1. Of course initially they cannot act with determination, they have to figure out randomly until they hit the alternating pattern, hence initially it has to be stochastic. The plan definitely works because we see that the population payoff average has reached the best possible. I suspect it remains like this till the end.

5.1.3 $\delta = 0.9$

Simulation 5

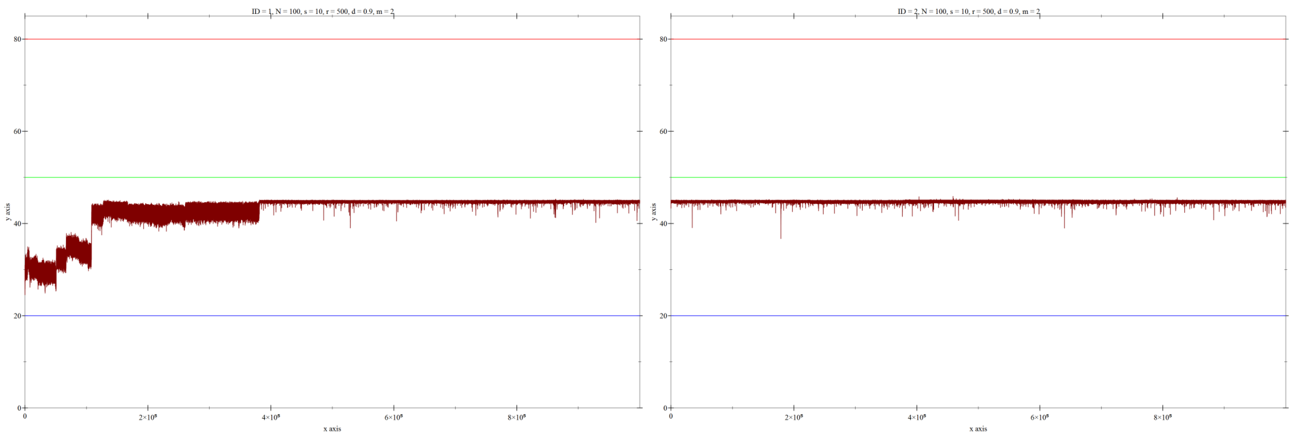

Figure 5.12: One typical run: $\delta = 0.9$

In this simulation, the tough strategy is back to stabilise the best possible outcome (the payoff sequence of 0 0 5 5 5...) we can ever hope for in a stochastic environment.

At 200000, there is this automaton:

```
(action 0.0 0.1)
 (list
  (list (action 0.3 0.4) (action 0.2 0.5) (action 0.7 0.2))
```

```
  (list (action 0.2 0.3) (action 0.0 1.0) (action 0.2 0.0))
  (list (action 0.2 0.0) (action 0.2 0.3) (action 0.0 0.9))))
```

Here it starts to play High with probability 0.9. After a HH the probability to play M is 0.9. After a MM, the probability to play Medium is 1. Medium is an absorbing state. They just need an initial period to figure it out. At 300000, the mechanism is fine tuned:

```
 (action 0.0 0.0)
 (list
  (list (action 0.5 0.5) (action 0.5 0.2) (action 0.4 0.2))
  (list (action 0.1 0.3) (action 0.0 1.0) (action 0.2 0.6))
  (list (action 0.7 0.2) (action 0.7 0.3) (action 0.0 0.9))))
```

The initial action scheme is to play High with probability 1 and after a HH to play Medium with probability 0.9, after a MM, the commitment to play Medium is absolute. Why do they play High in the first round? This is to repel always-medium. If it plays Medium in the first round, it would immediately be absorbed into the state of playing Medium, this would give always-medium the absolute payoff hence destabilising itself. This one also does not retreat to play Low after a HH, because if it retreats to play Low, it would give always-high a lot of points. It does not. It is a tough strategy.

At 400000 the mechanism is fine tuned further:

```
(action 0.0 0.0)
 (list
  (list (action 0.1 0.3) (action 0.8 0.0) (action 0.4 0.3))
  (list (action 0.1 0.3) (action 0.0 1.0) (action 0.3 0.2))
  (list (action 0.7 0.1) (action 0.7 0.3) (action 0.0 1.0))))
```

Initially it plays High with certainty, after a HH, it plays Medium with certainty, after a MM it plays Medium with certainty. Nothing else matters. This is the most direct way to get absorbed into the MM state that can still repel always-medium and always-high. This is evolution at its finest.

I suspect it stays like this till the end (including figure b).

5.2 Individuals with moderate patience

5.2.1 $\delta = 0.8$

Simulation 6

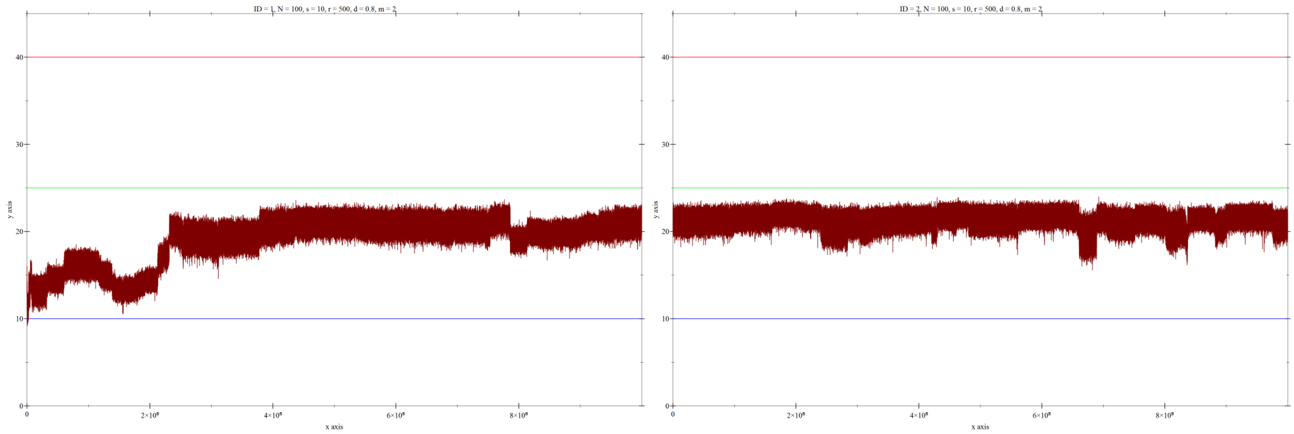

Figure 5.13: One typical run: $\delta = 0.8$

Simulation 7

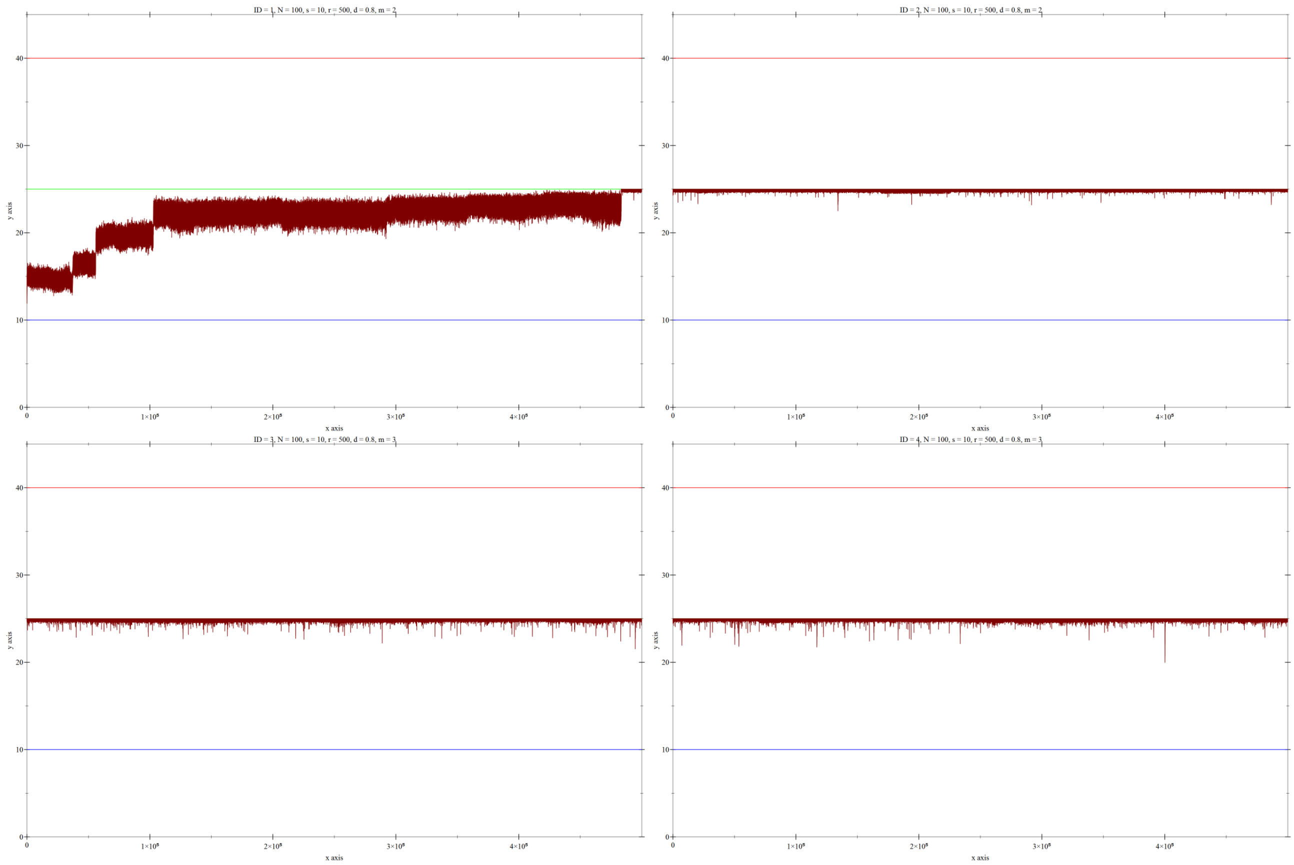

Figure 5.14: One typical run: $\delta = 0.80$

5.2.2 $\delta = 0.7$

Simulation 8

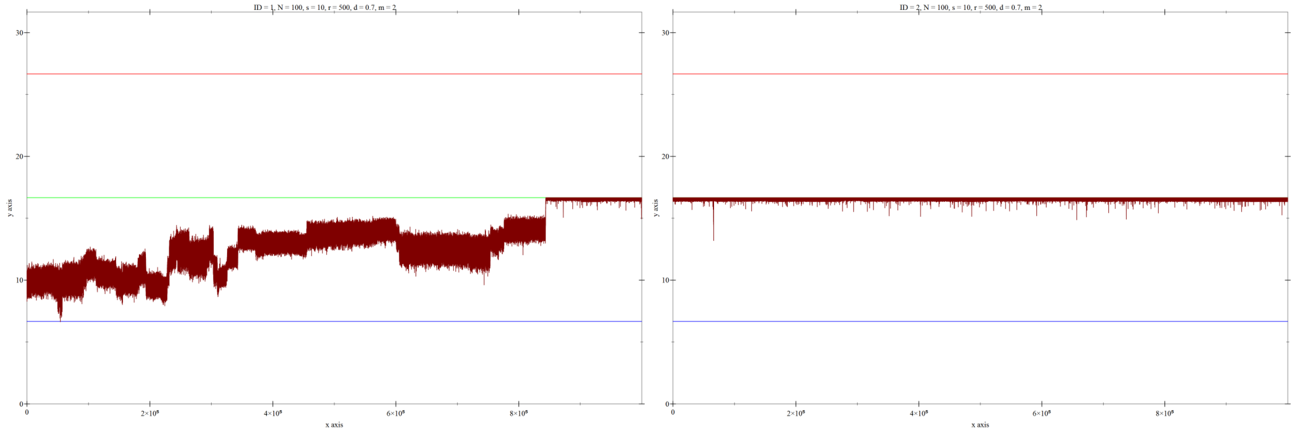

Figure 5.15: One typical run: $\delta = 0.7$

At 900000, the always-medium finds it way:

```
(action 0.0 1.0)
 (list
  (list (action 0.1 0.3) (action 0.0 0.0) (action 0.6 0.4))
  (list (action 0.0 0.9) (action 0.0 1.0) (action 0.0 0.9))
  (list (action 0.3 0.0) (action 0.1 0.8) (action 0.5 0.5))))
```

This one plays Medium initially with probability 1 and after MM, it commits absolutely to play Medium. Nothing else matters, it is an always-medium and the population average is the perfect one resulting from the perfectly equitable share.

5.3 Myopic players

5.3.1 $\delta = 0.5$

Simulation 9

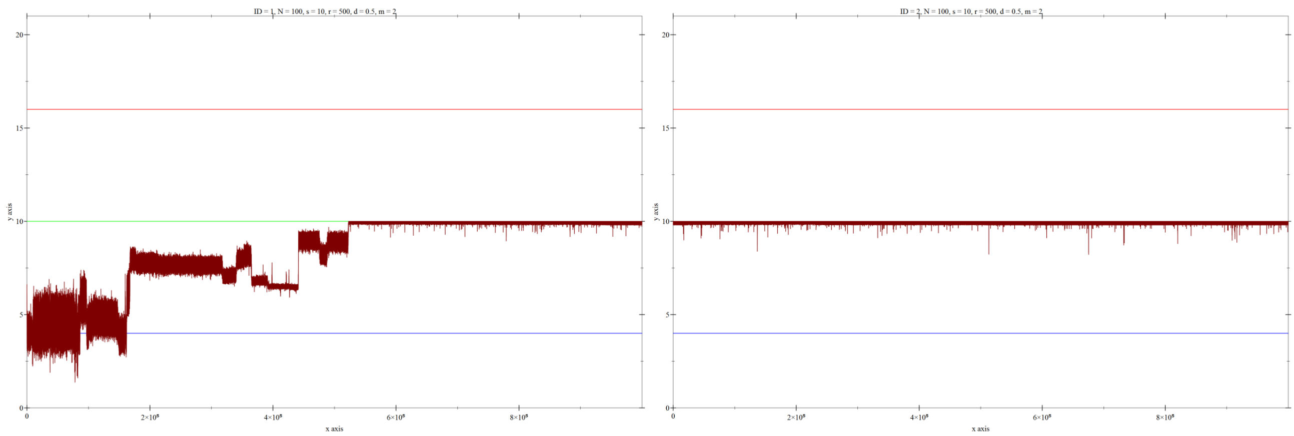

Figure 5.16: One typical run: $\delta = 0.5$

It is easier to reach the MM outcome because the vision is shorter, there is an aversion to wasting time and resources in the early rounds.

5.4 Conclusion

With highly patient players, in a stochastic world, the fully equitable share is virtually impossible. However, the population also does not see the chaotic mixture. With the markov chain in the strategy structure, there are two stable mechanisms to sustain the second-best possible outcome: a little less than ideal but still very high population payoff average.

- One is the tough strategy in which automaton spends some initial rounds (can be just one) to play aggressive and then settle down to play Medium forever after. This show of strength initially is crucial because it is devised to distinguish itself with always-medium and always-high. Among itself, it plays fair enough, with others, it repels completely. In this kind of structure, however, the probing can only happens initially (because after that it enters an absorbing state of playing Medium). In the finite state machine model, the probing period can happen later.

- The other is to use the initial phase to figure out a way to enter an absorbing pattern of alternating between 8 and 2. The faster the better. I wonder if this kind of mechanism has equivalent in a finite state machine model.

The third comment is that, as the patience level of the individuals homogeneously reduces, they inherently develops an aversion to wasting time and resources in the initial rounds. This affects the result a lot because the probe and retaliation can only happen initially. If people are obsessed with the present and near future payoff, no one would be willing to waste time on limited war. The only rational outcome is to surrender to peace immediately.

To briefly conclude, the perfectly equitable share maintained by an always-medium strategy has a mutation that leads directly to the collapse into the stable chaotic mix. And the way back takes two mutations: one to be fair to our own kind and one to fight off the aggressive type. More than that, the way back has a local maximum that is sustained by the tough negotiating mechanism: one that involves a series of probes, retaliation, strength demonstration, apology etc before the settling down of peace happens. This one, though less than ideal, is still something in better reach compared to a violently fluctuating state.

Anyway, the conclusion so far, if we are playing the long game, then limited war is justified on a prior ground. It sustains a local maximum that cushions the collapse of the negotiation process on the bargaining table. However, it also prevents the reaching toward the perfect maximum point of being all fair and square from the start. Stochastically speaking, it is even the only option that rules. After the period of probing and retaliation is over, then both parties can settle down into a sequence of playing Medium together or an alternating sequence of playing High and Low.

To go with the narrative, all the drama would not have happened if individuals are impatient. The lower the willingness to wait, the higher the price to pay to test each other initially.[1] Despite the encourage from referees that the simulation can have behavioral implications for lab experiments, due to the lack of knowledge from behavioral and experiment literature, I cannot give further details on a laboratory setup.

[1] At this point, I repeat the warning that taking interpretation from a simulated society to the outside world is not recommended.

Bibliography

[1] Aumann R. J. Agree to disagree. The annals of statistics 1976.

[2] Aumann, R. J. 1987b. Game theory. In The new Palgrave dictionary of economics, edited by M. Milgate and P. Newman, 460 - 482.

[3] Axelrod, R. (1980a). Effective choice in the prisoners dilemma. Journal of Conflict Resolution,24, pp. 3-25.

[4] Axelrod, R. (1980b) More effective choice in the prisoners dilemma. Journal of Conflict Resolution, 24, pp. 379-403.

[5] Axelrod, R. (1984). The Evolution of Cooperation. New York: Basic Books.

[6] Axelrod, R. (1987). Evolution of strategies in the iterated prisoners dilemma. In Genetic Algorithms and Simulated Annealing, L. Davis editor, Morgan Kaufman Publishers, Inc., pp. 32-41.

[7] Axelrod R.. *The evolution of cooperation: Revised edition*. New York: Basic Books, 2006.

[8] Axtell R.L., Epstein J.M., Young H.P.. The emergence of classes in a multi-agent bargaining model. In *Social Dynamics*. Washington: MIT Press, 2004.

[9] Binmore, K., A. Rubinstein, and A. Wolinsky. 1986. The Nash bargaining solution in economic modelling. Rand Journal of Economics 17: 176 - 188.

[10] Binmore K., Samuelson L., Young P., Equilibrium Selection in bargaining models. *Games and Economic Behavior*. 45, 296-328, 2003.

[11] Binmore K., Piccione M., Samuelson L., Evolutionary Stability in Alternating-Offers Bargaining Games. *Journal of Economic Theory*. 80, 257-291, 1998.

[12] Binmore K., Samuelson L., Evolutionary Drift and Equilibrium Selection. *Review of Economic Studies*. 66, 363-393, 1999.

[13] Boyd R., Lorberbaum J. P.: No pure strategy is evolutionarily stable in the repeated Prisoner's Dilemma game. *Nature*, 327, 58-59 (1987)

[14] Dawkins, R., 1983. Universal Darwinism. In: Bendall, D.S. (Ed.), Evolution from Molecules to Man. Cambridge University Press, Cambridge, pp. 403425.

[15] Danny Dolev, Dror G. Feitelson, Joseph Y. Halpern, Raz Kupferman, and Nati Linial. No justified complaints: On fair sharing of multiple resources. *Proceedings of 3rd Conference on Innovations in Theoretical Computer Science (ITCS 2012)*, 2012.

[16] Epstein J. M. Generative Social Science. Princeton and Oxford 2006.

[17] Hamilton W. D., 1967. Extraordinary Sex Ratios. Science 156: 477-488.

[18] Harsanyi, John C. 1956. Approaches to the bargaining problem before and after the theory of games: A critical discussion of Zeuthens, Hicks, and Nashs theories. Econometrica 24: 144 - 157.

[19] - 1963. A simplified bargaining model for the n-person cooperative game. International Economic Review 4: 194 - 220.

[20] Harsanyi, John C, and R. Selten. 1972. A generalized Nash solution for two-person bargaining games with incomplete information. Management Science 18: 80 - 106.

[21] - 1988. A general theory of equilibrium selection in games Cambridge: MIT Press.

[22] Hilbe C., Nowak M. A., Sigmund K., Evolution of extortion in Iterated Prisoner's Dilemma games. *PNAS*. 110, 6913-6918, 2013.

[23] Imhof LA, Fudenberg D, Nowak MA. 2005. Evolutionary cycles of cooperation and defection. PNAS 102: 10797-10800.

[24] Ian A. Kash, Eric J. Friedman, and Joseph Y. Halpern. Multiagent learning in large anonymous games. *Journal of AI Research 40*, 2011, pp. 571-598. A preliminary version appears in Proceedings of the Eighth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2009, pp. 765-772.

[25] Kreps D. M.. *A course in microeconomic theory*. Princeton University Press, 1990.

[26] Kuhn H W, *Lectures on theory of games*. Princeton. 1953.

[27] Lorberbaum J.: No strategy is evolutionarily stable in the repeated Prisoner's Dilemma. *J. theor. Biol.* 168, 117-130 (1994)

[28] Maynard Smith, J.. What use is sex?. Journal Theoretical Biology 30: 319 - 335.

[29] Maynard Smith, J.. *Evolution and Theory of Games*. Cambridge University Press, 1982.

[30] Maynard Smith J. and Price G.R. 1973. The logic of animal conflict. Nature 246: 1518. Maynard Smith J. and Szathmary E. 1995. The Major Transitions in Evolution. Oxford University Press, Oxford

[31] Myerson, R. B., Nash Equilibrium and the History of Economics Theory. *Journal of Economic Literature*. 37, 1067-1082, 1999.

[32] Myerson, R. B.. *Game theory: Analysis of conflicts*. Havard University Press, 1991.

[33] Nash Jr J F, Non-cooperative games, Ph.D. thesis, Mathematics Department, Princeton University, 1950.

[34] Nash Jr. J. F., The Bargaining Problem.. *Econometrica*. 18, 155-162, 1950.

[35] Nash, Jr. J. F., Noncooperative games. *Annals Math.*. 54, 289-95, 1951.

[36] Nash, Jr. J. F., Two-person cooperative games. *Econometrica*. 21: 128 - 140, 1953.

[37] Nowak M., Sigmund K., A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. *Nature*. 364, 56-58, 1993.

[38] Nowak MA, Sigmund K. 1992. Tit for tat in heterogeneous populations. Nature 355.

[39] Nowak MA, May RM. 1992. Evolutionary games and spatial chaos. Nature 359.

[40] Poza D, Galan JM, Santos JI, Paredes AL. An agent based model of the nash demand game in regular lattices. 2010.

[41] Press W. H., Dyson F. J., Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent. *PNAS*. 109, 10409-10413, 2012.

[42] Fogel, D. B. (1991) The evolution of intelligent decision making in gaming. Cybernetics and Systems, 22, pp. 223-236.

[43] Fogel, D. B. (1992). Evolving artificial intelligence. Doctoral Dissertation, University of California at San Diego.

[44] Fogel, D. B. (1993). Evolving behaviors in the iterated prisoners dilemma. Evolutionary Computation, 1 (1), pp 77-97.

[45] Fogel, L., A. Owens and M. Walsh (1966). Artificial intelligence through simulated evolution. New York: John Wiley & Sons.

[46] Fudenberg D, Tirole J. Game theory. MIT Press.

[47] Roth, A. E, Bargaining Experiments. In: *Handbook of Experimental Economics*, ed. John Kagel and Alvin E. Roth. Princeton University Press, 253-348, 1995.

[48] Rubinstein A., Perfect equilibrium in a bargaining model. *Econometrica*. 50, 97-109, 1982.

[49] Rubinstein, A., Z. Safra, and W. Thomson. 1992. On the interpretation of the Nash bargaining solution and its extension to non-expected utility preferences. Economettica 60: 1171 - 1186.

[50] Schelling, T. C.. *The Strategy of Conflict*. Havard University Press, 1980.

[51] van Veelen M., Garcia J., Rand D. G., Nowak M. A., Direct reciprocity in structured populations. *Proceedings of the National Academy of Sciences*. 109, 9929-9934, 2012.

[52] Vega-Redondo, F.. *Economics and Theory of Games*. Cambridge University Press, 2003.

[53] von Neumann J, Morgenstern O. Theory of Games and economics behavior. Princeton University Press.

[54] Weibul, J. W.. *Evolutionary Game Theory*. MIT Press, 1995.

[55] Young P.. *Inidividual Strategy and Social Structure*. Princeton University Press, 1998.