THE EFFECT OF SEGREGATION IN NON-REPEATED PRISONER'S DILEMMA

Thomas Nordli

University College of Southeast Norway

ABSTRACT

This article consolidates the idea that non-random pairing can promote the evolution of cooperation in a non-repeated version of the prisoner's dilemma. This idea is taken from[1], which presents experiments utilizing stochastic simulation. In the following it is shown how the results from [1] is reproducible by numerical analysis. It is also demonstrated that some unexplained findings in [1], is due to the methods used.

KEYWORDS

Evolution, cooperation, segregation, prisoner's dilemma.

1. INTRODUCTION

The problem of how cooperation can emerge in a population dominated by asocial behavior, is addressed by Axelrod and Hamilton in [2]. To answer this, they use clustering in a game of repeated prisoner's dilemma. It is called repeated because the players are given the possibility to play several rounds against the same opponent and the ability to remember the previous action of each opponent. This repetition together with a limited amount of memory give the players the opportunity to reciprocate. To model clustering they manipulate the probability of meeting an opponent with similar strategy the pairing is thus non-random. Axelrod and Hamilton found: Given conditions highly favoring cooperation in the long run, a low degree of clustering is sufficient for cooperation to survive. In [1] it were reported that several others (e.g. [3], [4], [5]) previously had been dealing with this idea, and that the work in [5] were the most similar.

The paper [1] goes further and argues that this may happen even in a non-repeating prisoner's dilemma where there are no possibilities for reciprocation. The players aka agents are given two possible strategies: always cooperate (applied by the cooperators) and always defect (applied by the defectors). By a simulated evolution, it is demonstrated that cooperation will prosper given certain initial parameter values. The parameters being (i) degree of clustering aka segregation, and (ii) initial fraction of cooperators. The paper also explores how variations on these parameters affect the dynamics of the simulated evolution and presents some findings of which some are unexplained.

One unexplained finding in [1] appears in an experiment presented as an initial benchmark without clustering. The share of cooperators starts with 90% but decreases fast almost linear. In the fourth generation, it reaches about 50% and the drop stops temporary. During the following

DOI:10.5121/ijcsit.2018.10201

generations — about ten — the share fluctuates around this level. This fluctuation period is called a plateau. After the plateau, the rapid decline continues until the total extinction of the cooperators.

A second unexplained finding appears in table 1 in [1] that presents different convergence ratios for cooperators and defectors on varying values on the two parameters segregation and initial fraction of cooperators. In this table three distinct regions appear: One region where no cooperators survive (region α), a second where both cooperators and defectors survive (region β) and a third where no defectors survive (region γ). It is remarked that a segregation parameter value of 0.5 (50%) works as a crossover point: To end up with cooperators exclusively, the initial segregation parameter value has to be set above this point (region γ). If the value is set below this, both types of players may sustain (region α and β). If the segregation parameter is set to this crossover point (or nearby), the simulation needs considerably more generations to converge. The paper [1] gives no explanation of neither the formation of these regions nor why the crossover point falls at 0.5.

The following two sections will introduce details from [1]: The model used (section 2. The Model) and the simulation algorithm that implements this model (section 3. The Simulation Algorithm). Section 4. The Numerical Analysis reproduces its results by numerical analysis. Then two sections follow dealing with the above-mentioned unexplained findings: The three regions (section 5. The three regions) and the choice of selection algorithm and its parameter setting (section 6. Changing a Parameter in the Algorithm). Finally a conclusion is made in section 7. Conclusion.

2. THE MODEL

This section explains the model used in [1].

The segregation is modelled as in [5] with one exception: The players are not given the ability to remember any previous meetings. Without memory, reciprocal strategies are not possible. Thus, only the two following strategies are used: (i) always cooperate and (ii) always defect. The model is based on a classical prisoner's dilemma where the players receive a payoff of either a, b, c or d where $a > b > c > d$. When a defector meets a cooperator, the defector gets a and the cooperator gets d. When two defectors meet, they both get c. And finally if two cooperators meet, each receive b. This is illustrated in table1.

Table 1: The payoffs in the Prisoner's Dilemma where a > b > c > d.

	cooperate	defect
cooperate	b, b	d, a
defect	a, d	c, c

2.1 THE PARAMETERS

Two parameters, proportion of cooperators (named p) and degree of randomness (named r) act on the model. Each of these parameters vary from 0 to 1. An overview of these parameters are shown in table 2.

The first parameter, proportion of cooperators in the population, is given as p, where $0 \leq p \leq 1, p \in \mathrm{R}$. Here 0 means that there are no cooperators and 1 means that the population consists exclusively of cooperators. As there are only one alternative strategy, namely defecting, the share of defectors will always be $1 - p$.

The second parameter is the degree of randomness in the process of matching pairs of agents. It is indicated by a parameter r, where $0 \leq r \leq 1, r \in \mathrm{R}$. This parameter is also called segregation parameter. When r is 0 there is no segregation, and when r is 1 there is full segregation. In full segregation, the meeting opponents always have the same strategy.

Table 2: The parameters of the model

Parameter	Description	Domain
p	Fraction of cooperators	$0 \leq p \leq 1, p \in \mathbb{R}$
r	Degree of randomness (aka segregation parameter)	$0 \leq r \leq 1, r \in \mathbb{R}$

2.2 MODELLING SEGREGATION

The probability of a cooperating agent meeting another cooperating agent is modelled as $r + (1-r)p$ and the probability of a defecting agent meeting another defecting agent as $r + (1 - r)(1 - p)$. If we need to calculate the probability of meeting an opponent with a different strategy one simply have to subtract these expressions from 1. This is summarized in in table 3.

Table 3: Modelling segregation

	Probability
(C meets C)	$r + (1 - r)p$
(D meets D)	$r + (1 - r)(1 - p)$
(C meets D)	$1 - (r + (1 - r)p)$
(D meets C)	$1 - (r + (1 - r)(1 - p))$

3. THE SIMULATION ALGORITHM

This section introduce the simulation algorithm presented in [1]. A refined algorithm, including details of the implementation necessary to understand the analysis done later in this document, is also included here. The evolution model is implemented as a stochastic agent based simulation. In this simulation each player is represented by an agent and the players' opponents are randomly picked (line 4 in algorithm 1).

The algorithm found in [1] is listed in algorithm 1. The choice of selection strategy is not documented in [1], but because I participated in the initiation of the project, I know that that truncation selection, with a truncation threshold of 0.5, was used. This was also confirmed (by personal communication) after publication (in 2013 2014). An extended version of the algorithm, including details of the selection algorithm, is shown in algorithm2.

Algorithm 1 as documented in [1].

```
 1: Initialize random population of players
 2: For all players: Set payoff =0
 3: while not stop do
 4:     Choose pairs for playing
 5:     For all players: Calculate payoff
 6:     Select players for surviving
 7:     Duplicate selected players with payoff>0
 8:     For all players: Set payoff =0
 9: end while
```

Algorithm 2 elaborating details on the selection strategy.

```
 1: Initialize random population of players
 2: For all players: Set payoff =0
 3: while not stop do
 4:     Choose pairs for playing
 5:     For all players: Calculate payoff
 6:     Sort the agents by payoff
 7:     Truncate the worst half of population
 8:     Replace the truncated players by duplicating the remaining ones, with payoff > 0.
 9:     For all players: Set payoff =0
10: end while
```

When using truncation selection, the individuals will be sorted by decreasing payoff. Duplicates of the players above the truncation threshold, given as s, will then replace those below. This is done by cloning the survivors $(\frac{1}{s} - 1)$ times, which in this case gives $(\frac{1}{0.5} - 1) = 1$time. Truncation selection is also described in [6] and [7].

As the truncation threshold used in [1] is 0.5, the population is cut in two equal sized halves, and the best half will be duplicated. The duplicates will replace the worst half. This is shown algorithmically in algorithm 2, where line 6 and 7 from algorithm 1 are replaced with line 6, 7 and 8 as seen in algorithm 2.

4. THE NUMERICAL ANALYSIS

The results presented in [1] where found by stochastic simulation. This section will reproduce these results by numerical analysis. The proportion of cooperators in a following generation will be estimated by a function of the two parameters, (p and r) as expressed in equation (1). The reproduction of the results appears when iterating this equation.

$$p^* = P\,(p,r) \tag{1}$$

4.1 ESTIMATING NUMBER OF SURVIVING COOPERATORS

The cooperators will be paid either *b* or *d*. Their proportion, *p*, can therefore be calculated as the sum of q_b and q_d, where q_b and q_d is the portion of population getting paid *b* and *d* respectively, as shown in equation (2):

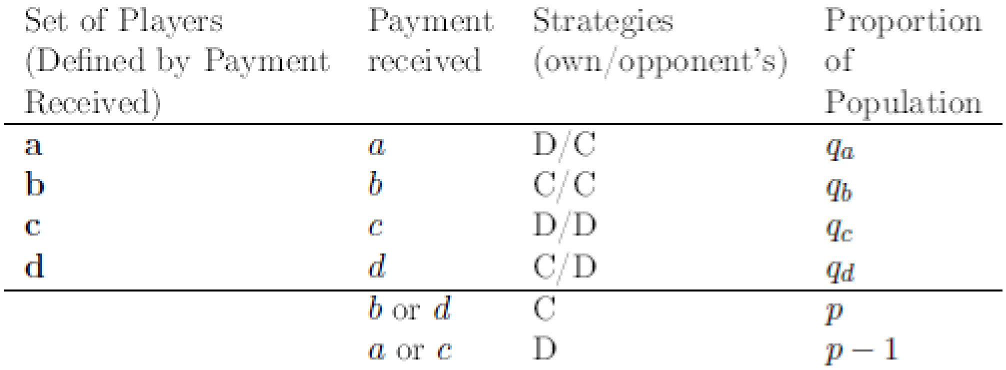
$$p = q_b + q_d \tag{2}$$

In the following let **a**, **b**, **c** and **d** denote the players that where paid *a*, *b*, *c* and *d* respectively. An overview of the notation used in this section, is given in table 4.

When all the payoffs are calculated for one generation, the agents are sorted by payoff (line 6 in algorithm 2). This will bring all of the **a**s to the top, followed first by the **b**s, then by the **c**s and the **d**s at the bottom. Thus, for any **d** to survive, q_d has to be greater than the truncation threshold.

Each time a defector is being paid *a*, a cooperator is paid *d*. The other way around is also true: When a cooperator gets *d*, a defector gets *a*. Intuitively we can see this as the probability of a cooperator meeting a defector is of course equal to the probability of a defector meeting a cooperator.

Table 4: Overview of notation used



Set of Players (Defined by Payment Received)	Payment received	Strategies (own/opponent's)	Proportion of Population
a	a	D/C	q_a
b	b	C/C	q_b
c	c	D/D	q_c
d	d	C/D	q_d
	b or d	C	p
	a or c	D	$p-1$

Since q_d and q_a always will be of equal size, they will always each be less or equal to 0.5, as shown in equation (3). This means that as long as the truncation threshold is 0.5 or greater, the **d**s will always stay below the threshold and none of them will ever survive to the next generation. The cooperators in the population will therefore exclusively consist of the **b**s and their clones.

$$0 \leq q_d = q_a \leq 0.5 \tag{3}$$

To estimate the number of cooperators after the truncation we must only consider the portion of q_b that will end up above the truncation threshold. There are two cases that needs to be calculated separately: (i) $(q_a + q_b) \leq \frac{1}{2}$ and (ii) $(q_a + q_b) > \frac{1}{2}$.

In the first case the better half will consist of **a**'s, **b**'s and **c**'s. Then all of the **b**'s should be counted. As each of them is duplicated, this number is doubled, giving us $2q_b$. In the second case the better half will consist only of **a**'s and **b**'s. If we subtract the **a**'s from the total number of agents in the upper half, we end up with the number of **b**'s. The difference $\frac{1}{2} - q_a$ will thus give the portion of **b**'s residing in the better half. This is multiplied by 2 to model the duplication, giving us $2(\frac{1}{2} - q_a)$. Equation (4) shows how the estimation of the share of cooperators in a forthcoming generation is modelled as a function of q_a and q_b.

$$p^* = \begin{cases} 2q_b, & (q_a + q_b) \leq \frac{1}{2} \\ 2\left(\frac{1}{2} - q_a\right), & (q_a + q_b) > \frac{1}{2} \end{cases} \tag{4}$$

4.2 ESTIMATING DISTRIBUTION OF PAYOFF

If we assume a big enough population, we can estimate how the different payments will be distributed in the population, based on the probabilities in the model. In addition we will assume that there is no restriction on how many times the agents can play in the same round. This is possibly a simplification and may not model exactly the way the simulation was implemented in [1], but the reproduction of their results indicates that such a deviation is negligible.

The probability of choosing a cooperator or a defector as the first agent is p and $1 - p$ respectively. To calculate the probability of the second agent (i_2) given the first (i_1), we have to multiply the probability of the first player with the second agent (given the first), the latter probabilities we take from the model in [1] (as shown in table 3). The probabilities are summarized in table 5.

Table 5: Probability distribution

	i_1	i_2	$i_2 \mid i_1$	Payoff to i_1
(D meets C)	$1-p$	$1-(r+(1-r)(1-p))$	$(1-p)(1-(r+(1-r)(1-p)))$	a
(C meets C)	p	$r+(1-r)p$	$(p)(r+(1-r)p)$	b
(D meets D)	$1-p$	$r+(1-r)(1-p)$	$(1-p)(r+(1-r)(1-p))$	c
(C meets D)	p	$1-(r+(1-r)p)$	$(p)(1-(r+(1-r)p)$	d

If we assume a big enough population, the probabilities will be equal to the distribution of payoff in the population. The proportion of players being paid a, b, c or d, denoted q_a, q_b, q_c or q_d respectively, may thus be calculated using the table 5.

We will to use these expressions to substitute q_a and q_b in equation (4) by their equivalents from table 5, but first we will simplify them. $(1 - p)(1 - (r + (1 - r)(1 - p)))$ is rewritten as $(1 - p)(1 - r)p$ and $(p)(r + (1 - r)p)$ becomes $p(r + (1 - r)p)$. This is summarized in table 6.

Table 6: Distribution of Payment in Non-random Environment

$$\begin{aligned} q_a &= (1-p)(1-r)p \\ q_b &= p(r+(1-r)p) \end{aligned}$$

After the substitution of q_a and q_b with the expressions from table 6 in equation (4), we get a new difference equation that is a function of p and r. This is shown in equation (5).

$$p^* = \begin{cases} 2p(r+(1-r)p), & p \leq \frac{1}{2} \\ 1-2p(1-r)(1-p), & p > \frac{1}{2} \end{cases} \tag{5}$$

4.3 ITERATING

As we are interested in the iterated version of the prisoner's dilemma, we use an algorithm that loops the difference equation (5), which is implemented as the function $P_C(p,r)$.

This function, $P_C(p,r)$, returns p^*. Initial values are set in the two parameters, p and r, for the first round of the loop round number 0. A variable named n is used to count the number of rounds. In the subsequent rounds, r is set to the value returned by P_C in the previous round. Only p will change during the run. This algorithm is repeated until p reaches 0 or 1. If this does not happens within *max* iterations ($n > max$), the proportion of cooperators have stopped changing from generation (round) to generation, and no further repetitions are done. This is presented in the lines 3 to 9 in algorithm 4.

4.4. REPRODUCING THE RESULT FROM [1]

To reproduce the results from [1] (reprinted as figure 1), algorithm 3 has to be run for all the combinations of p and r as they vary from 0.1 to 0.9 (with the step size of 0.1), as presented in algorithm 4.

The results of running this algorithm are shown in table 7. It clearly shows correspondence with the results of the simulation reported in [1]. In both the simulation and the calculation, three regions appear in the tables, as indicated in figure 2(a).

Algorithm 3 Iterating the prisoner's dilemma using numerical methods

```
$n \leftarrow 0$ {counting rounds, starting at 0}
$p_n \leftarrow$ some initial value
  $r \leftarrow$ some initial value repeat
    $p_{n+1} \leftarrow P_C(p_n, r)$ {equation (5)}

    $n \leftarrow n + 1$
  until $p = 0$ or $p = 1$ or $n >$ max

return
$p_n$
```

Algorithm 4 Reproducing the results from the simulation

```
1: for $p_0 = 0.1$ to 0.9 (step size: 0.1) do
2:      for $r = 0.1$ to 0.9 (step size: 0.1) do

3:           $t_{p0r} \leftarrow$ Algorithm 3($p_0, r$) {store resulting $p$ in table 7}

4:     end for

5: end for
```

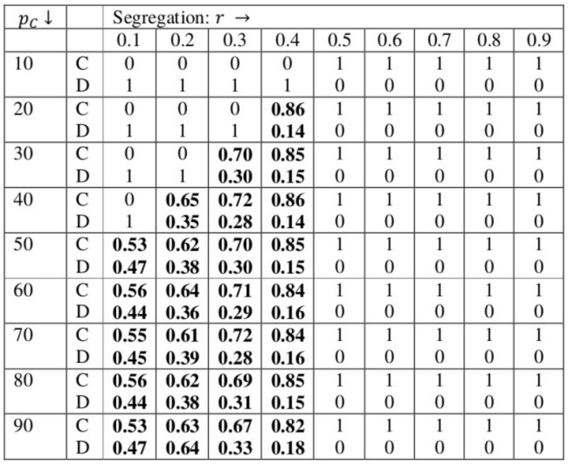
5. THE THREE REGIONS

This section will give an intuitive explanation of how three regions, α, $\beta$and γ, appears.

To get an intuitive understanding of why these three regions shown in figure 2(a) appear, we take a closer look at how the share of cooperators change from generation to generation. By comparing the values of p in two following generations, we get a value representing the growth of cooperators given p and r. We call this the growth rate of cooperators and define it as $p_{n+1} - p_n$. In table 8 the signs of these growth rates are shown.

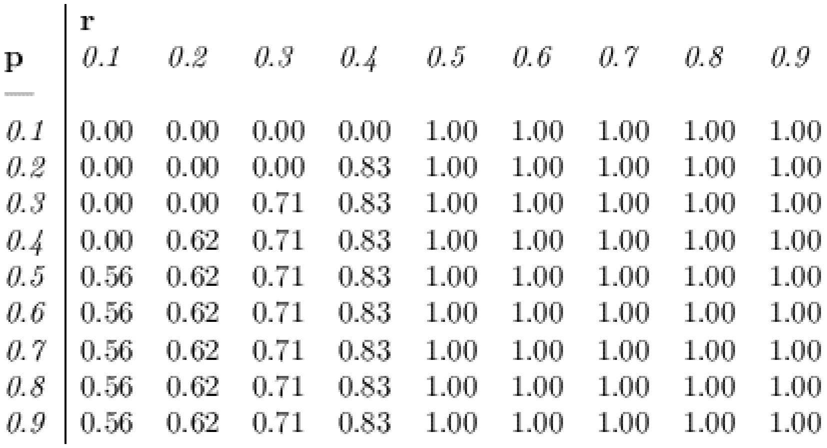
By looking at this table, it becomes clear how the three regions shown in figure 2(a) are formed.

Figure 1: Reprint of results presented in [1].

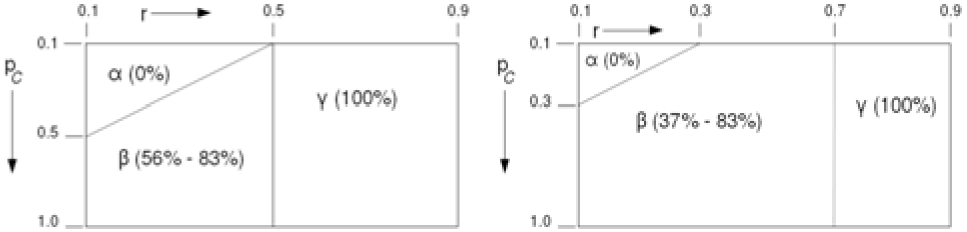


p_C ↓		Segregation: r →								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
10	C	0	0	0	0	1	1	1	1	1
	D	1	1	1	1	0	0	0	0	0
20	C	0	0	0	**0.86**	1	1	1	1	1
	D	1	1	1	**0.14**	0	0	0	0	0
30	C	0	0	**0.70**	**0.85**	1	1	1	1	1
	D	1	1	**0.30**	**0.15**	0	0	0	0	0
40	C	0	**0.65**	**0.72**	**0.86**	1	1	1	1	1
	D	1	**0.35**	**0.28**	**0.14**	0	0	0	0	0
50	C	**0.53**	**0.62**	**0.70**	**0.85**	1	1	1	1	1
	D	**0.47**	**0.38**	**0.30**	**0.15**	0	0	0	0	0
60	C	**0.56**	**0.64**	**0.71**	**0.84**	1	1	1	1	1
	D	**0.44**	**0.36**	**0.29**	**0.16**	0	0	0	0	0
70	C	**0.55**	**0.61**	**0.72**	**0.84**	1	1	1	1	1
	D	**0.45**	**0.39**	**0.28**	**0.16**	0	0	0	0	0
80	C	**0.56**	**0.62**	**0.69**	**0.85**	1	1	1	1	1
	D	**0.44**	**0.38**	**0.31**	**0.15**	0	0	0	0	0
90	C	**0.53**	**0.63**	**0.67**	**0.82**	1	1	1	1	1
	D	**0.47**	**0.64**	**0.33**	**0.18**	0	0	0	0	0

Table 7: Final proportion of cooperators as a function of r and p, with a truncation threshold of $\frac{1}{2}$



	r								
p	*0.1*	*0.2*	*0.3*	*0.4*	*0.5*	*0.6*	*0.7*	*0.8*	*0.9*
0.1	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00
0.2	0.00	0.00	0.00	0.83	1.00	1.00	1.00	1.00	1.00
0.3	0.00	0.00	0.71	0.83	1.00	1.00	1.00	1.00	1.00
0.4	0.00	0.62	0.71	0.83	1.00	1.00	1.00	1.00	1.00
0.5	0.56	0.62	0.71	0.83	1.00	1.00	1.00	1.00	1.00
0.6	0.56	0.62	0.71	0.83	1.00	1.00	1.00	1.00	1.00
0.7	0.56	0.62	0.71	0.83	1.00	1.00	1.00	1.00	1.00
0.8	0.56	0.62	0.71	0.83	1.00	1.00	1.00	1.00	1.00
0.9	0.56	0.62	0.71	0.83	1.00	1.00	1.00	1.00	1.00



(a) Truncation threshold: $\frac{1}{2}$ (b) Truncation threshold: $\frac{1}{3}$

Figure 2: Three regions, named *α*, *β* and *γ*, appears in the results

The area where the cooperators are driven to extinction, the corresponding area have negative growth. In the following, this region is referred to as area *α*.The players in this area will move upwards in the table from generation to generation, until *p* reach zero.

In addition, the second area *β*, is divided in two, one negative and one positive. The negative part is below the positive part. Players in the negative part will rise until they reach the positive part, getting positive growth rate, and switch direction. This again gives them a negative rate. In the stochastic simulation from [1] the rate will oscillate like this and converge to a stable situation containing both cooperators and defectors.

In the last area called *γ*, where the defectors are wiped out, the growth rate is positive. Here the players will continue downwards in the table until *p* is 1, and no defectors are left.

6. CHANGING A PARAMETER IN THE ALGORITHM

This section will investigate the effect of lowering the truncation threshold below the crossover point found in [1].

As indicated above, the choice of 0.5 as a truncation threshold results in that no d-s will survive from one generation to the other, in the numerical analysis.

The simulation done in [1] has a stochastic element which is absent from the numerical analysis which is completely deterministic. It is established above, that the numerical analysis reproduce the main results of the simulation in spite of the lack of randomness.

Choosing a truncation threshold below 0.5 will make some of the d-s survive even in the analysis. The equation is now changed so that only the better third of the population is considered and tripled. Everything else is the same. The modified equation is shown in equation (6).

$$P_C = \begin{cases} 3B, & A+B<\frac{1}{3} \\ 3\left(\frac{1}{3}-A\right), & (A+B\geq\frac{1}{3})\text{and}(A<\frac{1}{3}) \\ 0, & A\geq\frac{1}{3} \end{cases} \tag{6}$$

Table 8: Sign of growth rate as a function of *r* and *p*, with a truncation threshold of $\frac{1}{2}$.

	r								
p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	-	-	-	-	+	+	+	+	+
0.2	-	-	-	+	+	+	+	+	+
0.3	-	-	+	+	+	+	+	+	+
0.4	-	+	+	+	+	+	+	+	+
0.5	+	+	+	+	+	+	+	+	+
0.6	-	+	+	+	+	+	+	+	+
0.7	-	-	+	+	+	+	+	+	+
0.8	-	-	-	+	+	+	+	+	+
0.9	-	-	-	-	+	+	+	+	+

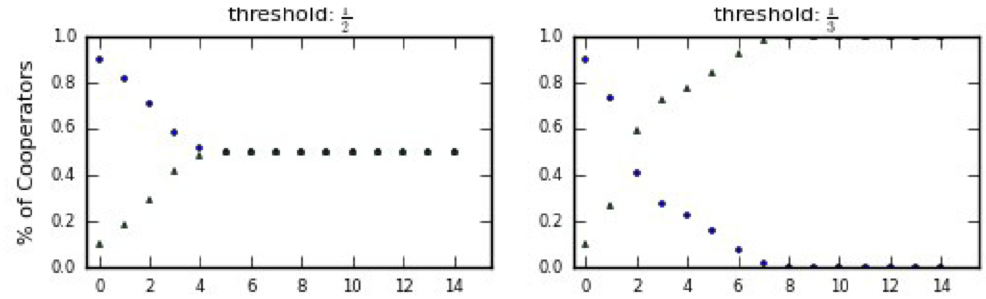
Table 9: Final proportion of cooperators as a function of *r* and *p*, with a truncation threshold of $\frac{1}{3}$

	r								
p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	0.00	0.00	0.48	0.56	0.67	0.83	1.00	1.00	1.00
0.2	0.00	0.42	0.48	0.56	0.67	0.83	1.00	1.00	1.00
0.3	0.37	0.42	0.48	0.56	0.67	0.83	1.00	1.00	1.00
0.4	0.37	0.42	0.48	0.56	0.67	0.83	1.00	1.00	1.00
0.5	0.37	0.42	0.48	0.56	0.67	0.83	1.00	1.00	1.00
0.6	0.37	0.42	0.48	0.56	0.67	0.83	1.00	1.00	1.00
0.7	0.37	0.42	0.48	0.56	0.67	0.83	1.00	1.00	1.00
0.8	0.37	0.42	0.48	0.56	0.67	0.83	1.00	1.00	1.00
0.9	0.37	0.42	0.48	0.56	0.67	0.83	1.00	1.00	1.00

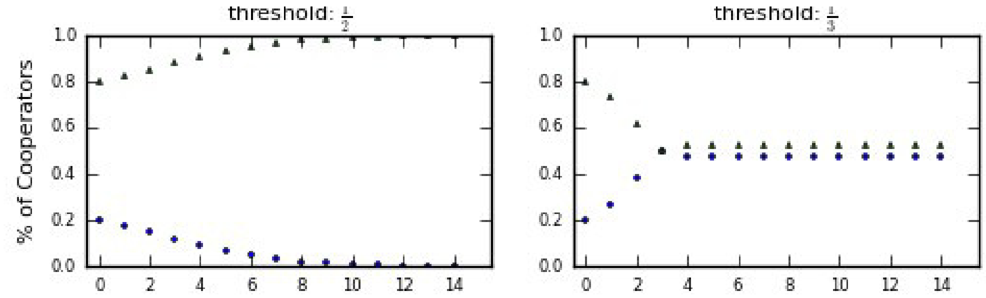
6.7 TRUNCATION THRESHOLD : $\frac{1}{2}$ VS .$\frac{1}{3}$

Figure 3(a) corresponds to the benchmark case without cooperation. This corresponds to figure 3 in [1]. Even when starting with 90% cooperators, they are quickly exterminated. We see that when using a truncation threshold of $\frac{1}{2}$, the plateau is present, just as in [1], but absent when the threshold is set to $\frac{1}{3}$.

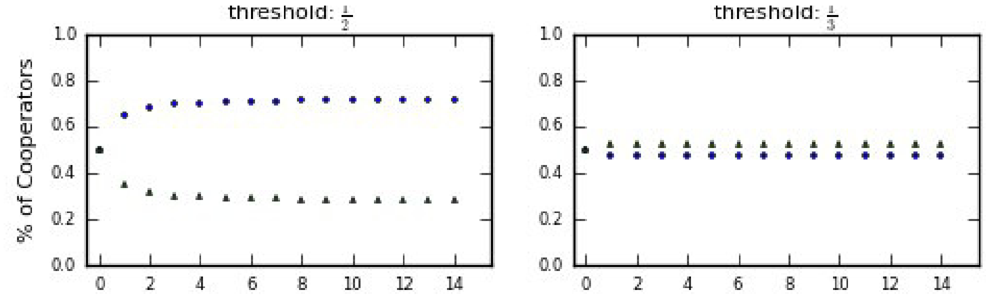
The figures 3(b), 3(c) and 3(d), corresponds to the figures 4, 5 and 6 in [1] by having the same values on *r* and *p*. Where the truncation threshold is 0.5, the result of the numerical analysis matches the stochastic simulation. Setting the threshold to $\frac{1}{3}$ makes changes to the three regions, as shown in figure 2(b), making all three of end up in area β.



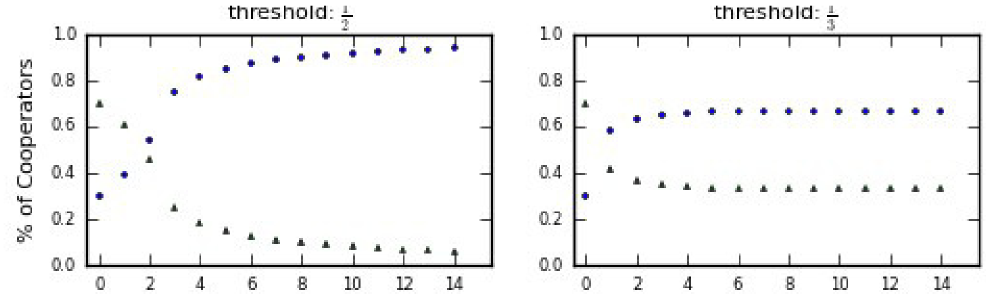
(a) Numerical analysis with *r* =0.0, and p=0.9 (corresponds to figure 3 in [1].)



(b) Numerical analysis with *r* =0.3, and p=0.2 (corresponds to figure 4 in [1].)



(c) Numerical analysis with *r* =0.3, and p=0.5 (corresponds to figure 5 in [1].)



(d) Numerical analysis with r =0.5, and p=0.3 (corresponds to figure 6 in [1].)

Figure 3: Visualizing the result of changing the truncation threshold.

Table 10: Sign of growth rate as a function of *r* and *p*, with a truncation threshold of $\frac{1}{3}$.

	r								
p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	-	-	+	+	+	+	+	+	+
0.2	-	+	+	+	+	+	+	+	+
0.3	+	+	+	+	+	+	+	+	+
0.4	-	+	+	+	+	+	+	+	+
0.5	-	-	-	+	+	+	+	+	+
0.6	-	-	-	-	+	+	+	+	+
0.7	-	-	-	-	-	+	+	+	+
0.8	-	-	-	-	-	+	+	+	+
0.9	-	-	-	-	-	-	+	+	+

7. CONCLUSION

This paper support the conclusion from [1] that segregation plays a considerable role in evolution of cooperation in a game of Prisoner's Dilemma, in spite of the absence of repeated action and reciprocity. However, it argues that the choice of truncation selection as selection strategy together with a truncation threshold of 0.5 is the cause of the plateau and crossover point from [1]. It also explains the formation of the three regions.

REFERENCES

[1] N. B. Jon Reiersen, Matching structure and the evolution of cooperation in the prisoner's dilemma, International journal of computer science & information technology (IJCSIT), vol. 5, pp. 109 119, Aug. 2013.

[2] R. Axelrod and W. D. Hamilton, The evolution of cooperation, Science, vol. 211, no. 4489, pp. 1390 1396, 1981.

[3] W. D. Hamilton, The genetical evolution of social behaviour. ii, Journal of theoretical biology, vol. 7, no. 1, pp. 17 52, 1964.

[4] M. A. Nowak, Five rules for the evolution of cooperation, science, vol. 314, no. 5805, pp. 1560 1563, 2006.

[5] R. Boyd and P. J. Richerson, The evolution of reciprocity in sizable groups, Journal of theoretical Biology, vol. 132, no. 3, pp. 337 356, 1988.

[6] D. Thierens and D. Goldberg, Convergence models of genetic algorithm selection schemes, in Parallel Problem Solving from Nature PPSN III (Y. Davidor, H.-P. Schwefel, and R. M nner, eds.), vol. 866 of Lecture Notes in Computer Science, pp. 119 129, Springer Berlin Heidelberg, 1994.

[7] H. M hlenbein and D. Schlierkamp-Voosen, Predictive models for the breeder genetic algorithm i. continuous parameter optimization, Evolutionary computation, vol. 1, no. 1, pp. 25 49, 1993.