Carlotta Bonsignori, Gianpaolo Caramellino, Marco Ghiani

Axelrod 2.0

description using the ODD protocol

1 Purpose

The aim of our program is the simulation of a simplification of the Axelrod Tournament. Axelrod is an American political scientist that organized a tournament in which each participant designed a strategy to be used in a game in which the Prisoner Dilemma was iterated 200 times. Despite the complexity of some of the strategies, it turned out that the winner of the tournament was the Tit-For-Tat strategy, that cooperates on the first shot and then replicates what the opponent did in the previous round. To sum up, Axelrod concluded that a good strategy for the finitely repeated Prisoner Dilemma should have some features: being nice (never be the first to defect), being provocable (return defection when the other player defeated, cooperation when the other chose cooperate), avoid being envious (do not cheat your partner) and avoid being too clever (namely, avoid too complex actions) pays off.

2 State variables

The tournament is played by sixteen players characterized by the low-level state variables described in Table 1:

Name of the variable	**Description of the variable**	**Default value**
who	Identification number	Randomly assigned natural number in [0, 15]
defect?	Describes the chosen action: if agent defects it takes value true, otherwise false	false
partner_defected?	Describes what each of the player's opponents played in the last meeting	List of sixteen false
partner_defected_before?	Describes what each of the player's opponents played two meetings ago	List of sixteen false
payoff	Tells the total payoff accumulated by the player	0

Table 1

The tournament can be repeated from one up to 200 times. The environment can be setted to be neutral (baseline world where the proportion of players playing a given strategy is the same for all strategies), cooperative (the share of individuals playing a cooperative strategy is sensibly higher) or selfish (the share of individuals playing a defecting strategy is dominant).

3 Process overview and scheduling

In the tournament every player has to play against all the others and every round consists in repeating the static Prisoner Dilemma a number of times. The number of times, that we made a choice variable, is a factor that characterizes the classification of the strategies.

First, one player is selected randomly through the instruction `ask players` in procedure `start-meetings`; the selected player is asked to choose a strategy according to her own identification number; also all the players with `who`-number larger than the `who` of the selected player are asked to choose an action. Then each player is asked to update `partenr_defected?` and `partner_defected_before?` according to the move taken by her partners.

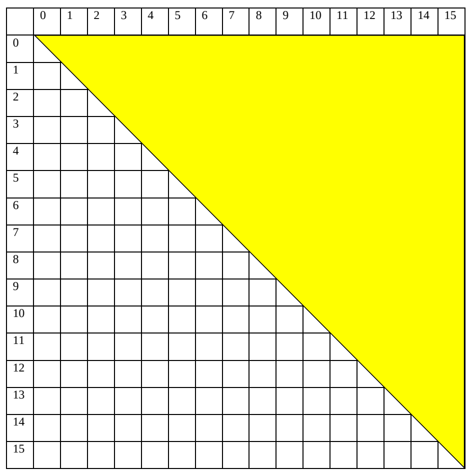


Figure 1

By asking each player to meet only the players with a who higher than her own who we are sure that in a tournment each player plays exactly one time with each other player according to a triangular scheme (Figure 1). The game is the classical Prisoner Dilemma, where we assume that payoffs are collected according to the following payoff matrix:

	C	D
C	3, 3	0, 5
D	5, 0	1, 1

Figure 2

A schematic representation of the schedule of a tournament is offered in Figure 3.

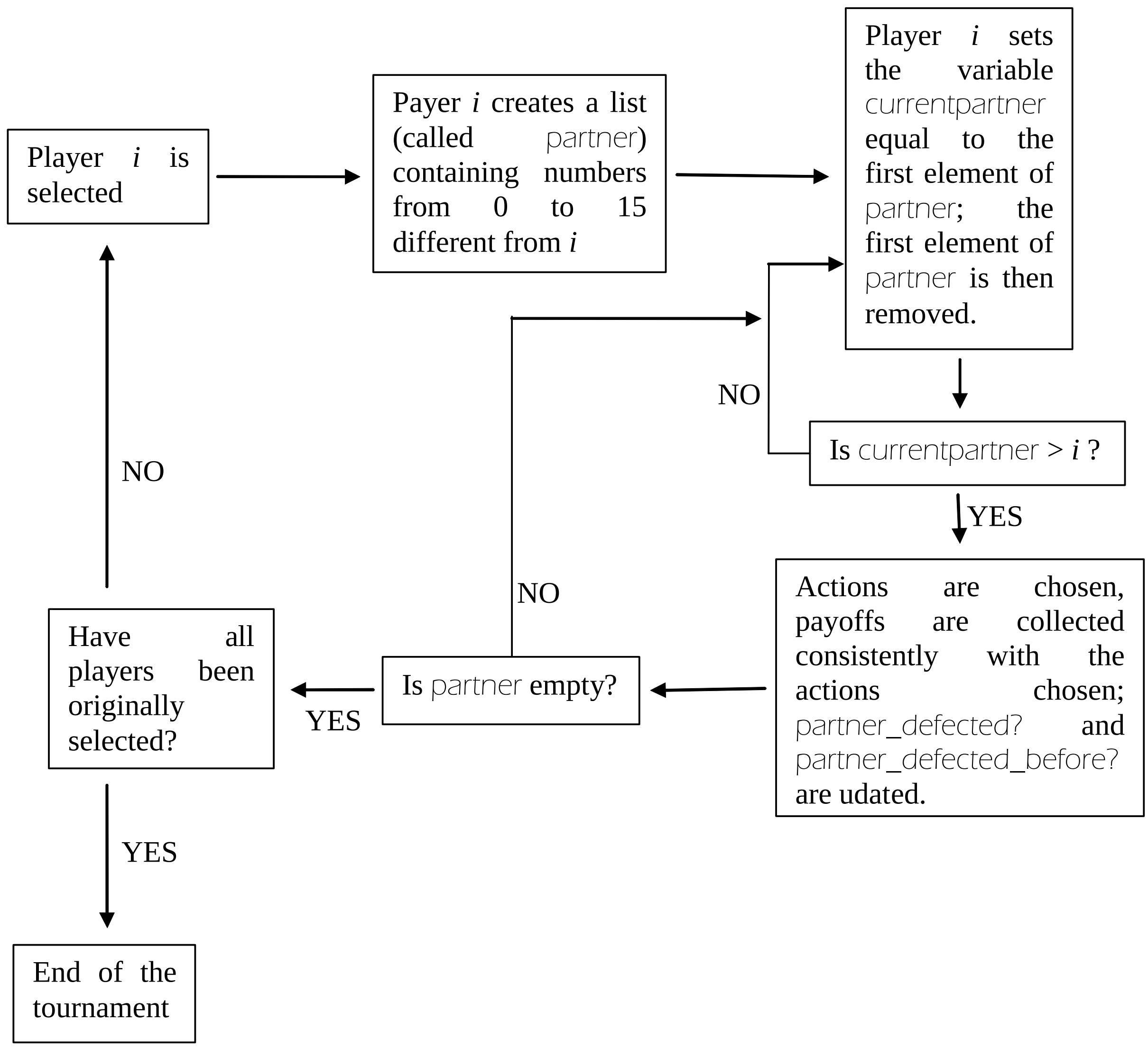

Player i is selected
Payer i creates a list (called partner) containing numbers from 0 to 15 different from i
Player i sets the variable currentpartner equal to the first element of partner; the first element of partner is then removed.
NO
Is currentpartner > i ?
YES
Actions are chosen, payoffs are collected consistently with the actions chosen; partner_defected? and partner_defected_before? are udated.
NO
Is partner empty?
YES
Have all players been originally selected?
NO
YES
End of the tournament

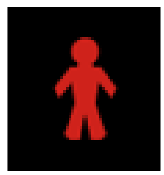
Figure 3

4 Design Concepts

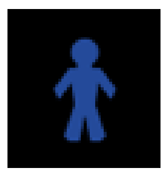
4.1 Emergence

The dynamics of the entire repeated game emerge from the individual behavior and its interaction with the environment; each individual follows a specific strategy within an environment that can be setted to be neutral, cooperative or selfish. The strategies defined in the model are:

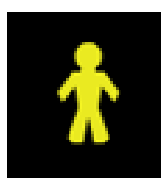
- always C: the player will always play C, no matter what the past history of the game is (i.e., no matter what she or her competitor played the previous shot);



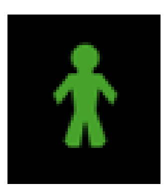
- always D: the player will always play D, no matter what;



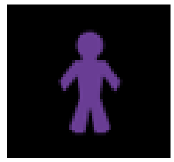
- Tit-for-Tat: the player will initially choose C and then, after seeing what the competitor chose in the last shot, she will play the opponent's previous action;



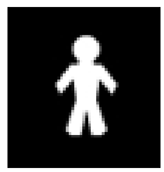
- Tit-for-2Tat: the structure is the same of Tit-for-Tat; what differs is that the player will play D only after 2 consecutive Ds of the competitor, so she is less easily provocable;



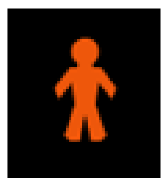
- 2Tit-for-Tat: the structure is the same of Tit-for-Tat; what differs is that the player will again play C (after he played D) only after 2 consecutive Cs of the competitor, so she is less forgiving;



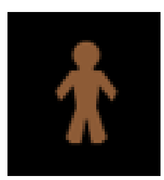
- random 1: the player will play $\frac{1}{2}$of the times C and $\frac{1}{2}$ D;



- random 2: the player will play $\frac{8}{10}$ of the times C and $\frac{2}{10}$ D;



- random 3: the player will play $\frac{2}{10}$ of the times C and $\frac{8}{10}$D



The results might be strongly conditioned by the chosen environment and by the number of iterations of the tournament. It must be also stressed that the payoff structure is a crucial determinant for the result of our simulation: by fixing the payoff structure presented in Figure 2 we will find that certain strategies will behave better than others. If the payoffs would have been fixed differently, the tournament would have led to different classifications of the strategies.

4.2 Adaptation

Individuals play specific strategies and are not allowed to change them; the different strategies vary a lot in complexity: in particular some strategies display an high adaptiveness (e.g the tit-for-tat) and some others don't (e.g. the one which prescribes to cooperate always). Actually one of the aims of this work is to show how and in which measure a more adaptive behavior can influence welfare in different environments.

4.3 Fitness

Individuals collect payoffs which is the measure of fitness in the model. Players (and therefore strategies) are classified by evaluating the average payoffs they get in the entire tournament. Payoff is collected according to the payoff matrix in Figure 2.

4.4 Sensing

Each player does not know, before she moves, which action has been chosen by the other player. After each player has selected an action, the payoff of the game (and therefore the opponent's action) is revealed. The observed action is memorized by the agent up to two matches: that is, each agent is able to recollect the action played by each of her opponents the last and the second last time they met. No information from the environment is given to the players: i.e. the world in which the game is played is not a relevant variable in the decision making process.

4.5 Interaction

Since the payoffs that the two players get depend both on their own action and on the other player's strategy, game theorists have developed a concept of rationality in order to include strategic behaviours, namely to include behaviours that anticipate or reply to particular actions of the opponent. In this simple game, given the payoff structure, the theory suggests that there is an incentive to cheat for both players so that in the one shot Prisoner Dilemma the equilibrium turns out to be Pareto inefficient. The impossibility to reach a Pareto efficient solution of the game, has led game theorists to reconsider the concept of rationality to be able to explain cooperative behaviour of people. Under certain conditions, they found that cooperation is perfectly rational when the game is infinitely repeated. In the context of repeated games, a strategy is the multi period correspondence to the concept of action in the one shot game. To be more precise a strategy is a complete plan of action, one for every period the player is called to move.

4.6 Stochasticity

Stochasticity is introduced through the definition of three random strategies. The introduction of this stochasticity element avoids the ranking of the strategies to be deterministically determined; in particular it appears that if the tournament is repeated a few times, the results can be significantly influenced by the presence of the random strategies.

4.7 Observation

Individual average payoffs are registered; since to each individual corresponds a specific strategy, each strategy is associated with a determined average payoff. The strategies are ranked according to the average payoff they yield. The observed results are summarized in Table 2.

World		Baseline		Cooperative		Selfish
Winner		2Tit-for-Tat		Defect		2Tit-for-Tat
Looser		Cooperate		Cooperate		Cooperate
Players	AvP_Coop_1	2.01	AvP_Coop_1	2.5	AvP_Coop_1	0.89
	AvP_Def_1	2.35	AvP_Def_1	3.82	AvP_Def_1	1.66
	AvP_Tit-for-Tat_1	2.4	AvP_Tit-for-Tat_1	2.7	AvP_Tit-for-Tat_1	1.65
	AvP_Tit-for-2Tat_1	2.3	AvP_Tit-for-2Tat_1	2.64	AvP_Tit-for-2Tat_1	1.58
	AvP_2Tit-for-Tat_1	2.53	AvP_2Tit-for-Tat_1	2.77	AvP_2Tit-for-Tat_1	1.71
	AvP_random1_1	2.24	AvP_random1_1	3.15	AvP_random1_1	1.32
	AvP_random2_1	2.15	AvP_random2_1	2.8	AvP_random2_1	1.07
	AvP_random3_1	2.31	AvP_random3_1	3.61	AvP_random3_1	1.56
	AvP_Coop_2	1.99	AvP_Coop_2	2.49	AvP_Def_2	1.67
	AvP_Def_2	2.34	AvP_Coop_3	2.51	AvP_Def_3	1.65
	AvP_Tit-for-Tat_2	2.41	AvP_Coop_4	2.49	AvP_Def_4	1.66
	AvP_Tit-for-2Tat_2	2.29	AvP_Coop_5	2.51	AvP_Def_5	1.67
	AvP_2Tit-for-Tat_2	2.53	AvP_Coop_6	2.49	AvP_Def_6	1.67
	AvP_random1_2	2.21	AvP_Coop_7	2.49	AvP_Def_7	1.68
	AvP_random2_2	2.17	AvP_Coop_8	2.52	AvP_Def_8	1.69
	AvP_random3_2	2.35	AvP_Coop_9	2.49	AvP_Def_9	1.64
Average		2.28625		2.74875		1.548125
Max		2.53		3.82		1.71
Min		1.99		2.49		0.89

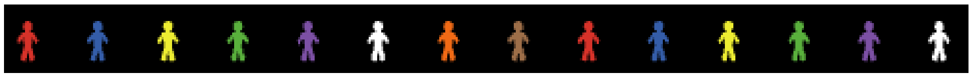
Table 2

The striking result is that the strategy cooperate is the one losing in all three possible worlds, however the Cooperative World is the one in which the average payoffs are significantly higher then in the other two worlds. In fact the average of the average payoff s in the Cooperative World is 2.75, while in the Baseline World it is 2.29 and in the Selfish World it is 1.55. Notice that the average payoff of the worst strategy in the Cooperative World, 2.49, is much higher than the average payoff of the best strategy in the Selfish World, 1.71, and just slightly lower than the average payoff of the best strategy in the Baseline World, 2.53.

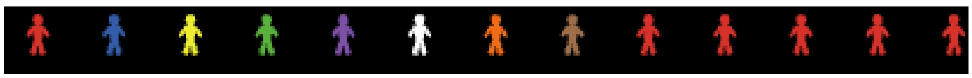
5 Input

The environmental conditions that impose a dynamic of certain state variables are called inputs. They are summarized for the three possible worlds in what follows:

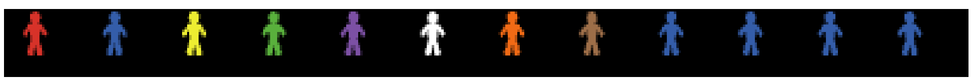
- Baseline world: the 16 players are paired up with the 8 different strategies, so that there is a couple of players for each strategy;



- Cooperative world: in this world there are 9 players that play the strategy always C and the other 7 players are endowed each one with one of the remaining 7 strategies;



- Defective world: in this world there are 9 players that play the strategy always D and the other 7 players are endowed each one with one of the remaining 7 strategies



The default values of the state variables have been already summarized in the section "State Variables".

6 Submodels

In this section we will discuss the procedures of the model more in detail.

1. `start-meetings`: The procedure `start-meetings` is the core of the program since it lets players meet, it makes each player collect her own payoff and it stores the relevant information in players memory. First, one player is selected randomly; this player is asked to create a list containing the `who` values of all the other (unselected) players, sorted randomly; this list is called `partner`. Once the list `partner` is created, the procedure asks the selected player to assign to the specifically created variable `currentpartner` the value of the first `who` stored in `partner`. Then, for reasons that will be clearer afterwards, we eliminate the first element of the list partner. There are now two possibilities: either the initially selected agent plays with the player whose `who` is equal to `currentpartner` or she doesnt. In order to be sure that each player meets all the others, playing only one match with each one of her opponents in each tournment, we decide to let each player play only with the opponents having

an higher `who` than she has. In this way we are sure that all the players play the same number of games and that eachone meets everybody exactly one time before the tournment ends. So the procedure asks to the selected player to compare her own `who` with the value she previously assigned to `currentpartner`; if her own `who` is higher the procedure asks her to go back to the list partner (that meanwhile has been emptied of its first element) and, anew, attribute to the variable `currentpartner` the value of the first element of the updated list. If instead, the `who` of the selected player is lower than the `who` assigned to `currentpartner`, then the meeting starts: agents choose their actions, collect their payoffs, memorize what each player they met played and then another player can be selected to repeat all the commands until all players have executed the commands in the block.

2. `choose_action:` The procedure asks the player to choose her strategy according to who she is and according to the world that has been set in the Chooser.

3. `get_payoff:` Each player is asked to execute the procedure payoff. In this procedure the player is asked to check both his action (inspetting the content of her own `defect?`) and her opponent's action (checcking how the player whose `who` is equal to the value of `currentpartner` setted her `defect?`). Then the value of `payoff` is updated consistently with the payoff matrix.