UNIVERSITÉ LIBRE DE BRUXELLES ULB

Learning Dynamics

INFO-F409

Assignment 2

28 November 2021

TAILLANDY Valentin

ULB student ID : 000542194

UNIVERSITÉ **L**IBRE DE **B**RUXELLES (ULB)

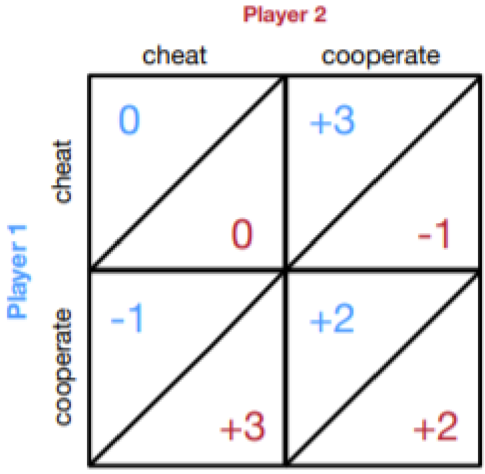
Contents

Part 1: Evolutionary dynamics in infinite populations

We have the three following strategies:

- Always cooperate
- Always cheat
- Copycat

Always cooperate and always cheat strategies are self-explanatory, Copycat will play cooperate and will then copy whatever the other player does. They will play R rounds with the following payoff matrix (figure 1):

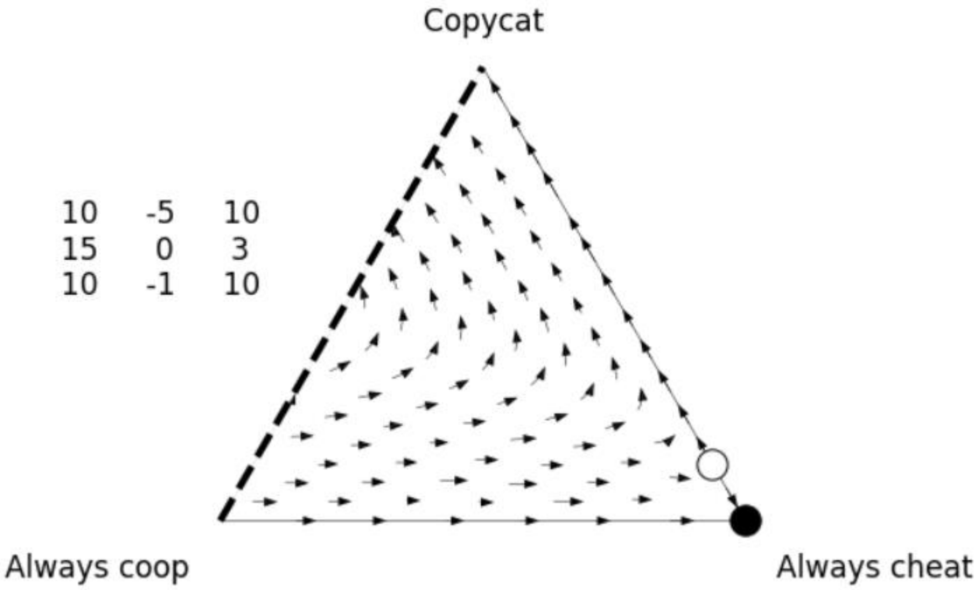


(Figure 1)

The expected payoff for these strategies against each other is as follows:

	Always cooperate	Always cheat	Copycat
Always cooperate	2R/2R	-R/3R	2R/2R
Always cheat	3R/-R	0/0	3/-1
Copycat	2R/2R	-1/3	2R/2R

R=5



a)

As we can see there is one unstable equilibrium between Copycat and Always cheat, Copycat and Always cheat are bistable.

There is 3 stable equilibrium one in each corner, Always cooperate, Always cheat and Copycat but because they represent the homogeneous population of that strategy.

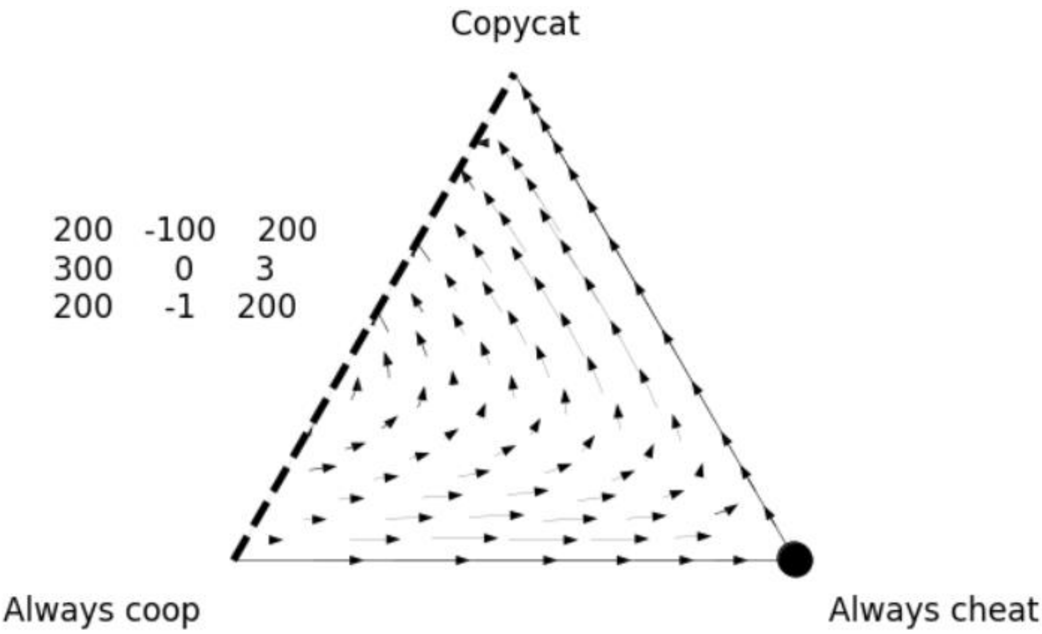
We do have a line of stable equilibrium between Copycat and Always cooperate which seems logical because Copycat when playing against an Always cooperate can be viewed as an Always cooperate strategy because he will first cooperate then copy the always cooperate resultingly with him always cooperating.

b)

- Always cheat dominates Always cooperate.
- Copycat and Always cheat are bistable, so they will dominate each other depending on their proportion.
- Copycat and Always cooperate coexist as no strategy dominate the other.

We can conclude that only Always cheat dominate Always cooperate.

R=100



a)

Here, there is no unstable equilibrium; we still have the 3-corner stable equilibrium representing the homogenous population and the line of stable equilibrium between Copycat and Always cooperate.

b)

- Always cheat dominates Always cooperate.
- Copycat dominates Always cheat.
- Always cooperate and Copycat coexist.

We can conclude that Always cheat dominates Always cooperate, we weren't expecting a change here as the matchup is one-sided but now, we have Copycat that dominate Always cheat as Copycat get a better payoff in the long term as R grew.

Part 2: Evolutionary dynamics in finite populations

We have the five following strategies :

- Always cooperate (Cooperator)
- Always cheat (Cheater)
- Copycat
- Grudger
- Detective

Figure 2. Description of the strategies that you should study in this assignment (image taken from https://ncase.me/trust).

First, we want to calculate the fixation probabilities between each 2 monomorphic states, therefore, we need to compute the payoff matrix of how each strategy fare against each other's. Then we will compute the fixation probabilities from these payoff matrices.

Notes:

- R is assumed to be equal to 100.
- Population = 100
- Beta = 1
- Mu → 0

We are taking the expected payoff, so over the R rounds.

We will modify theses value later (except mu), but for the purpose of the explanation we will produce results for this specific case:

	Always cooperate	Always cheat
Always cooperate	2/2	-1/3
Always cheat	3/-1	0/0

```
Coop invade Cheater with p = 0.0
Cheater invade Coop with p = 0.6431012069684492
```

We can see that a Cheater will invade the population of Cooperator with a probability of ~64%.

	Always cooperate	Copycat
Always cooperate	2/2	2/2
Copycat	2/2	2/2

```
Coop invade Copycat with p = 0.01
Copycat invade Coop with p = 0.01
```

We can see that a Cooperator or a Copycat will not invade the other's population because the probability isn't greater than 1/100 = 1% (the random drift).

	Copycat	Always cheat
Copycat	2/2	-0.01/0.03
Always cheat	0.03/-0.01	0/0

```
Copycat invade Cheater with p = 0.09546966745184281
Cheater invade Copycat with p = 0.0
```

We can see that a Copycat will invade the population of Cheater with a probability of ~9,5%.

	Copycat	Grudger
Copycat	2/2	2/2
Grudger	2/2	2/2

```
Copy invade Grudger with p = 0.01
Grudger invade Copycat with p = 0.01
```

We can see that a Copycat or a Grudger will not invade the other's population because the probability isn't greater than 1/100 = 1% (the random drift).

	Always cheat	Grudger
Always cheat	0/0	0.03/-0.01
Grudger	-0.01/0.03	2/2

```
Cheater invade Grudger with p = 0.0
Grudger invade Cheater with p = 0.09546966745184281
```

We can see that a Grudger will invade the population of Cheater with a probability of ~9,5%.

	Always cooperate	Grudger
Always cooperate	2/2	2/2
Grudger	2/2	2/2

```
Coop invade Grudger with p = 0.01
Grudger invade Coop with p = 0.01
```

We can see that a Cooperator or a Grudger will not invade the other's population because the probability isn't greater than 1/100 = 1% (the random drift).

	Always cooperate	Detective
Always cooperate	2/2	-0.91/2.97
Detective	2.97/-0.91	1.98/1.98

```
Coop invade Detective with p = 0.0
Detective invade Coop with p =  0.6358402195177939
```

We can see that a Detective will invade the population of Cooperator with a probability of ~63%.

	Always cheat	Detective
Always cheat	0/0	0.09/-0.03
Detective	-0.03/0.09	1.98/1.98

```
Cheater invade Detective with p = 0.0
Detective invade Cheater with p = 0.08304808537364716
```

We can see that a Detective will invade the population of Cheater with a probability of ~8,3%.

	Copycat	Detective
Copycat	2/2	1.98/1,98
Detective	1.98/1,98	1.98/1,98

```
Copycat invade Detective with p = 0.013310998242380903
Detective invade Copycat with p = 0.004995765377157003
```

We can see that a Copycat will invade the population of Detective with a probability of ~1,3% but a Detective will not invade the population of Copycat as the probability is below the random drift.

	Grudger	Detective
Grudger	2/2	0.07/0.03
Detective	0.03/0.07	1.98/1.98

```
Grudger invade Detective with p = 0.0
Detective invade Grudger with p = 0.0
```

We can see that a Grudger or a Detective will not invade the other's population because the probability is equal to 0 for both.

For our game with the five presented strategy, the expected payoff matrix is:

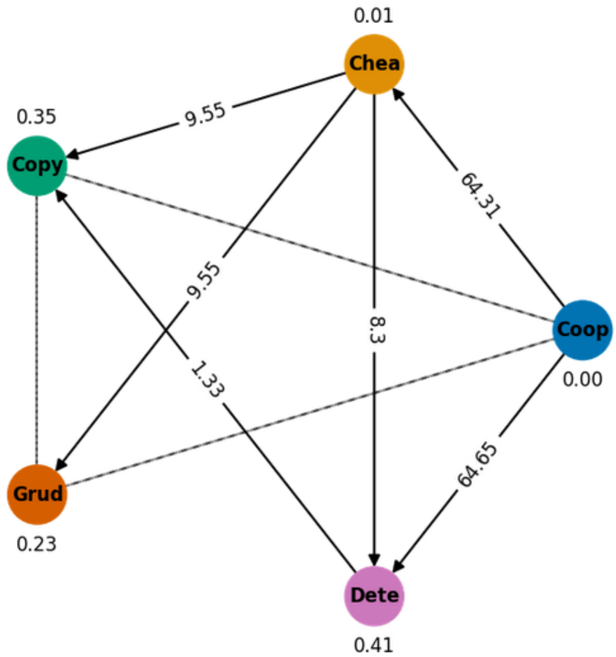
	Always cooperate	Always cheat	Copycat	Grudger	Detective
Always cooperate	2/2	-1/3	2/2	2/2	-0.91/2.97
Always cheat	3/-1	0/0	0.03/-0.01	0.03/-0.01	0.09/-0.03
Copycat	2/2	-0.01/0.03	2/2	2/2	1.98/1.98
Grudger	2/2	-0.01/0.03	2/2	2/2	0.07/0.03
Detective	2.97/-0.91	-0.03/0.09	1.98/1.98	0.03/0.07	1.98/1.98

We can then compute the transition matrix of the reduced Markov Chain formed by the 5 strategies:

```
[[0.67260275 0.         0.0025     0.0025     0.         ]
 [0.1607753  0.93150314 0.         0.         0.         ]
 [0.0025     0.02386742 0.99375106 0.0025     0.00332775]
 [0.0025     0.02386742 0.0025     0.995      0.         ]
 [0.16162195 0.02076202 0.00124894 0.         0.99667225]]
```

```
transition probabilities (Cooperator Cheater Copycat Grudger Detective and read as row invading column)
```

And finally, plot the invasion diagram:



We can see that:

-Copycat dominates {Cheater, Detective} and is not dominated.

-Grudger dominates {Cheater} and is not dominated.

-Detective dominates {Cooperator, Cheater} and is dominated by {Copycat}.

-Cooperator dominates no strategy and is dominated by {Detective, Cheater}.

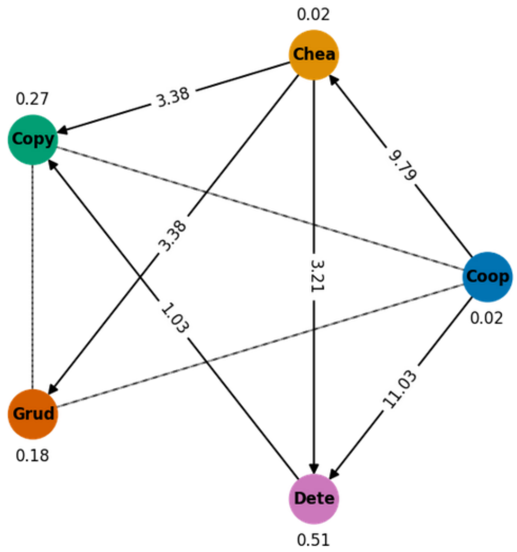
-Cheater dominates {Cooperator} and is dominated by {Grudger, Copycat, Detective}.

a) Effect of β (intensity of selection)

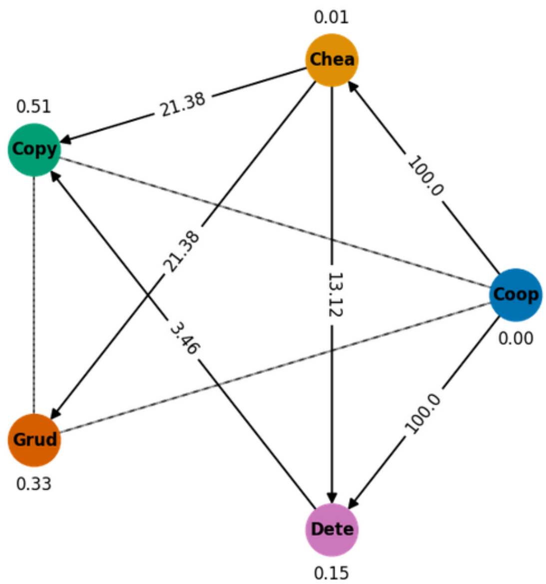
Beta, the intensity of selection represents how much we are going to select the individuals that are performing best regarding their fitness. For example, a beta equal to 10 puts much more emphasis on the individual's fitness, meaning the ones that are performing best (highest fitness) will be selected, and we are more likely to imitate the best performing strategies. But with a beta of 0.1 it will keep the fitness close together and the individual's fitness will not have such a big impact meaning we are more likely to explore than copy.

(Z = 100)

$\beta = 0.1$:



$\beta = 10$:

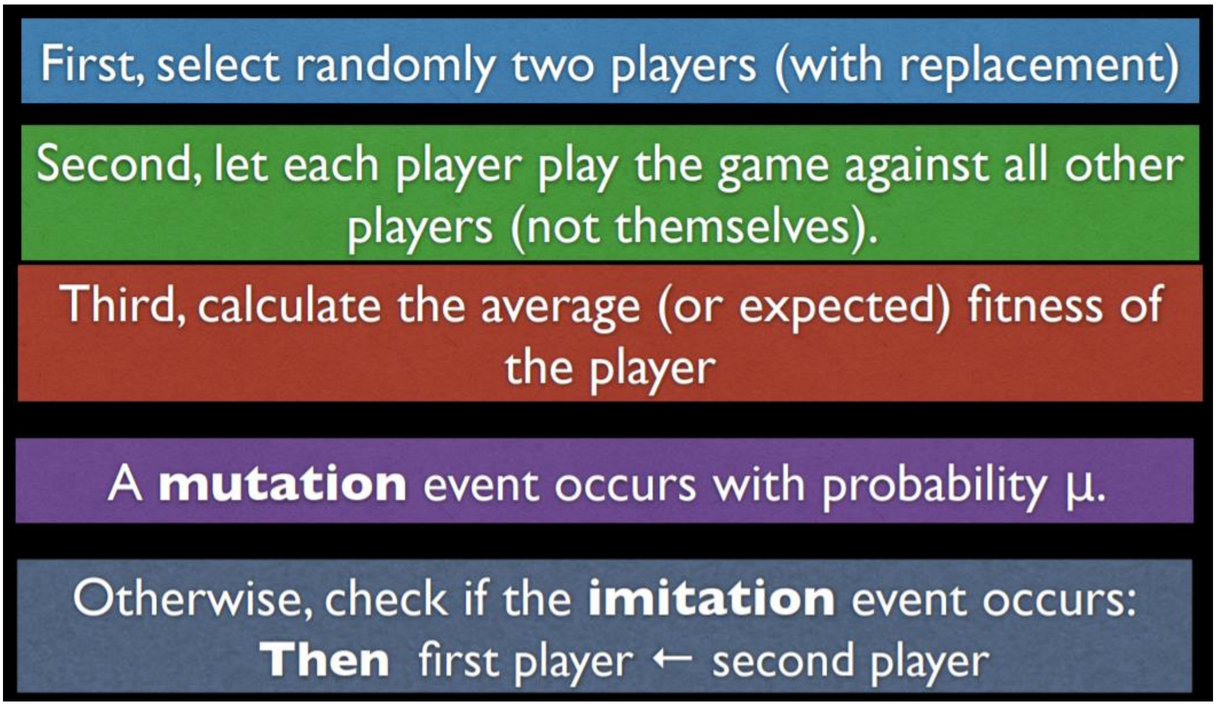


With β=0.1 to β=10 the fixation probabilities increase, for example an Always Cooperate population will now get invaded by a Detective or a Cheater with a probability of 100%, meaning they have no chance of not getting invaded which seems logical as any differences in fitness will be amplified. Furthermore, as said previously with a large β, we are concentrating more on individuals' fitness and not exploration, we can see that with the stationary distribution. With β=10, we have a smaller proportion of Detective (a "medium" performing strategy compared to Grudger or Copycat) compared to β=0.1, this throws us back to the fact that the slightest difference in fitness will have an influence on the selection. "Medium" performing strategy are eclipsed by the "best" strategy and "bad" strategy disappear completely.

In conclusion we should be careful when choosing a beta because it could polarize the strategy if too big or the reverse, not give enough advantages to one's fitness if too small.

b) Effect of Z (population size)

The process of social learning is done as follows:

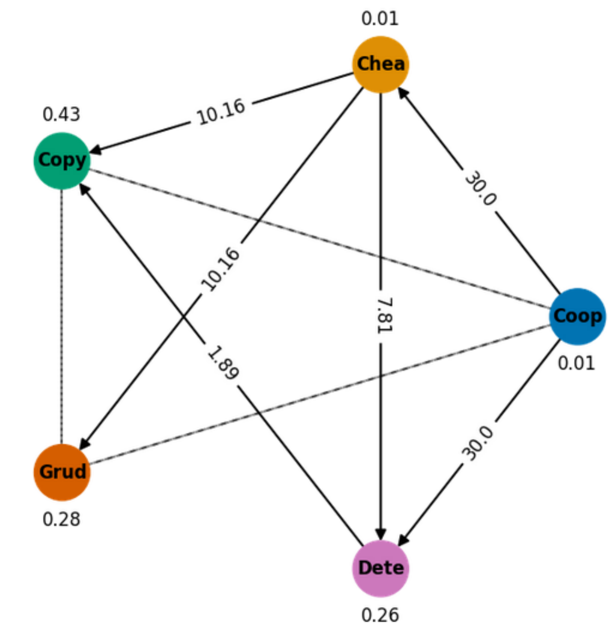


[1]

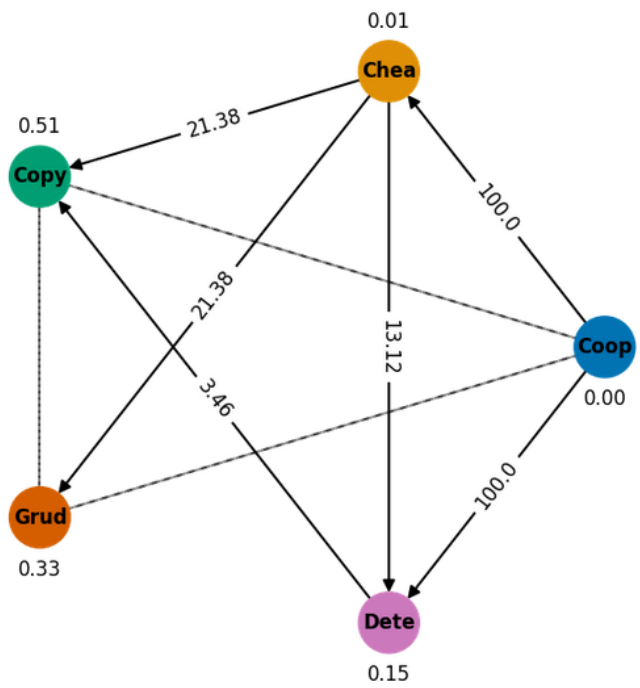
This means that, when we are doing the imitation process, the player will play against the rest of the population, the more the population grows the more the differences in average payoff grows.

(beta = 10)

Z=30:



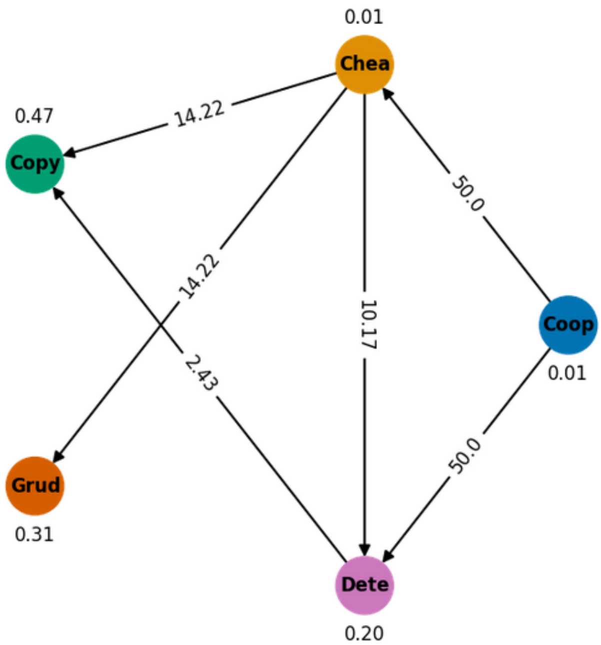
Z=100:



We can see that the stationary distribution has changed, the more the population sizes increase, the harder it gets for weaker strategy (dominated by a lot of other strategy) to face the increased number "predator" (strategy dominating the weaker strategy). They are getting a bad matchup more often.

c) Special case where β=10 and Z=50

$\beta = 10$ and Z = 50:



In our settings, they are no dominant strategy per say, Copycat and Grudger dominates strategy and are not dominated by any strategy, but they are not dominating all strategy because they are coexisting. Therefore, no strategy is dominant in our settings.

Part 3: Monte-Carlo simulations

Here, we want to estimate the stationary distribution through Monte-Carlo simulations. For this we need to create two functions:

moran_step(current_state, beta, mu, Z, A) -> next_state

and

estimate_stationary_distribution(nb_runs, transitory, nb_generations, beta, mu, Z, A) -> stationary_distribution

We have the parameters as follow:

$$\beta = 10, \mu = 1e^{-3}, Z = 50, R = 100$$

$$transitory = 10^3, nb\_generations = 10^5, nb\_runs = 10$$

The code has been commented and for the sake of readability, the lines of code will not be explained here, only the goal of each method. Please check *moran_process.py* for more information.

select_random_without_replacement(population, number):

This function returns "number" randomly selected player out of a "population" with their position in the population and their strategy. Note that they are selected without replacement, meaning no one can be selected twice in the same select.

estimate_fitness(selected, population, Z, A):

This function returns the fitness of both "selected" player in regard to the "population". They will play the "population" (themselves not included) and we will calculate the fitness with the "A" average payoff matrix and divide by the "Z" population size.

def prob_imitation(beta, fitness):

This function uses the fermi function to estimate the probability that player 1 imitates player 2 strategies.

$$p = \left[1 + e^{\beta(f(1) - f(2))}\right]^{-1}$$

[2]

The function f(1) representing the fitness of player 1 or f(2) the fitness of player 2.

moran_step(current_state, beta, mu, Z, A):

This function will return the next_state of the population regarding its "current_state", "β", "μ", the population size "Z" and the average payoff matrix "A". The algorithm is present in "Part 2: Evolutionary dynamics in finite populations, section b)".

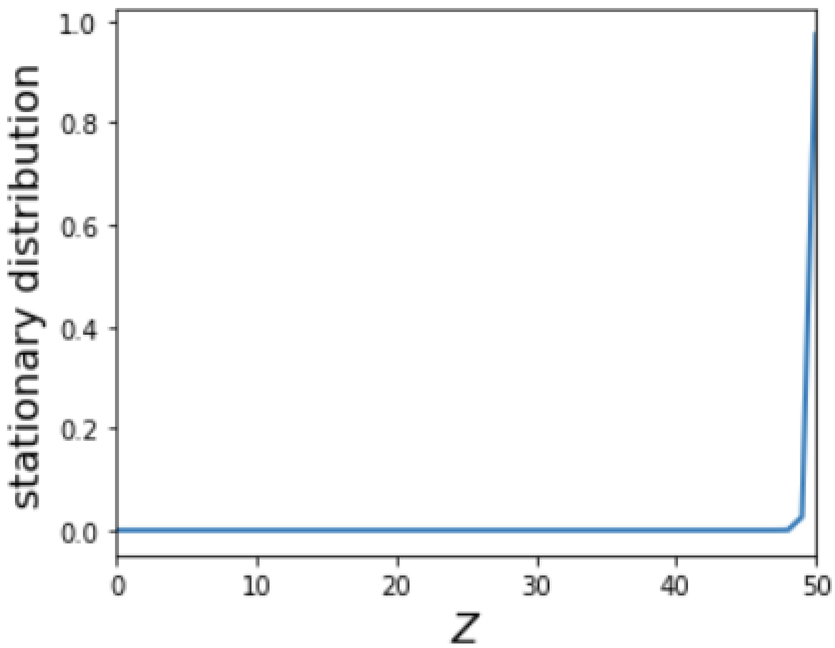
estimate_stationary_distribution(nb_runs, transitory, nb_generations, beta, mu, Z, A):

This function will return the stationary distribution as a vector of floats; this will contain the fraction of time the population has spend in a state. For this we will proceed as follow:

- Start from a random population state.
- Run the Moran process for a transitory period of "transitory" generations.
- Afterwards, run the process for "nb_generations" generations and count how often the population passes through each possible state.
- Repeat the simulation 10 times and average the results.

[3]

With all these functions and the parameters, we have the stationary distribution (y-axis) for each possible number of Copycat strategies (x-axis) in the population:



Acknowledgments

[4] *EGTTools: Toolbox for Evolutionary Game Theory* has been used and of great help during the part 1 and part 2. For the 2-dimensional simplex, the fixations probability between each two monomorphic states, the transition matrix of the reduced Markov Chain formed by the five monomorphic states and the resulting invasion diagram.

References

[1] Nowé, T. L. (n.d.). *part 5 EGT and cooperation-allbuild.*

[2] Nowé, T. L. (n.d.). *part 5 EGT and cooperation-allbuild.*

[3] Fernández Domingos, E. (n.d.). *Assignment 2*

[4] Fernández Domingos, E. (2020). *EGTTools,* Retrieved from https://github.com/Socrats/EGTTools