

Progress report: Learning vol2

Madis Ollikainen

June 26, 2015

1 New/updated learning schema

I updated the learning schema such that after every time step each possible strategy for each player gets an updated value, based on the gain the player would have gotten if they would have played a specific strategy. The info about the value of each strategy is held in an array of doubles called *memory*, which is of length $N_{agents} \times N_{strategies}$. All entries in *memory* are initialised to 1. After each round t each agents i updates its memory entries M_{ij} for the all strategies j . The update $M_{ij}(t) \rightarrow M_{ij}(t+1)$ is done as

$$M_{ij}(t+1) = M_{ij}(t) \left(1 + \xi \frac{gain_{ij}}{w_i(t-1)} \right),$$

where $w_i(t-1)$ is the wealth of agent i before round t and ξ is a pre-defined learning factor, which can be set in the configuration file under the *mUpDate* variable. The $gain_{ij}$ is calculated from the knowledge of what other players did during the time step t . That mean, that after time step t each player evaluates how well he would have done, if he would have chosen any of the strategies, keeping the choice of the other player fix to the ones they actually did.

Before each round the strategies of all agents are updated (! *the actual strategy being played* !) using the entries in the array memory array M . The updating schema has the same probabilistic structure as it had in the previous simulations. The probabilities of each strategy j for agent i are calculated as

$$p_{ij} = \frac{e^{\beta M_{ij}}}{\sum_k e^{\beta M_{ik}}}.$$

Test result for $\xi = \{0.1, 0.05, 0.01\}$ using $N = 1200$ players during $T = 50$ time steps are shown in the graphs below. A smaller ξ value seems to slow the process of the αr ordering schema giving an unequal society. *I'll run some longer simulations to see if this truly is like that.*

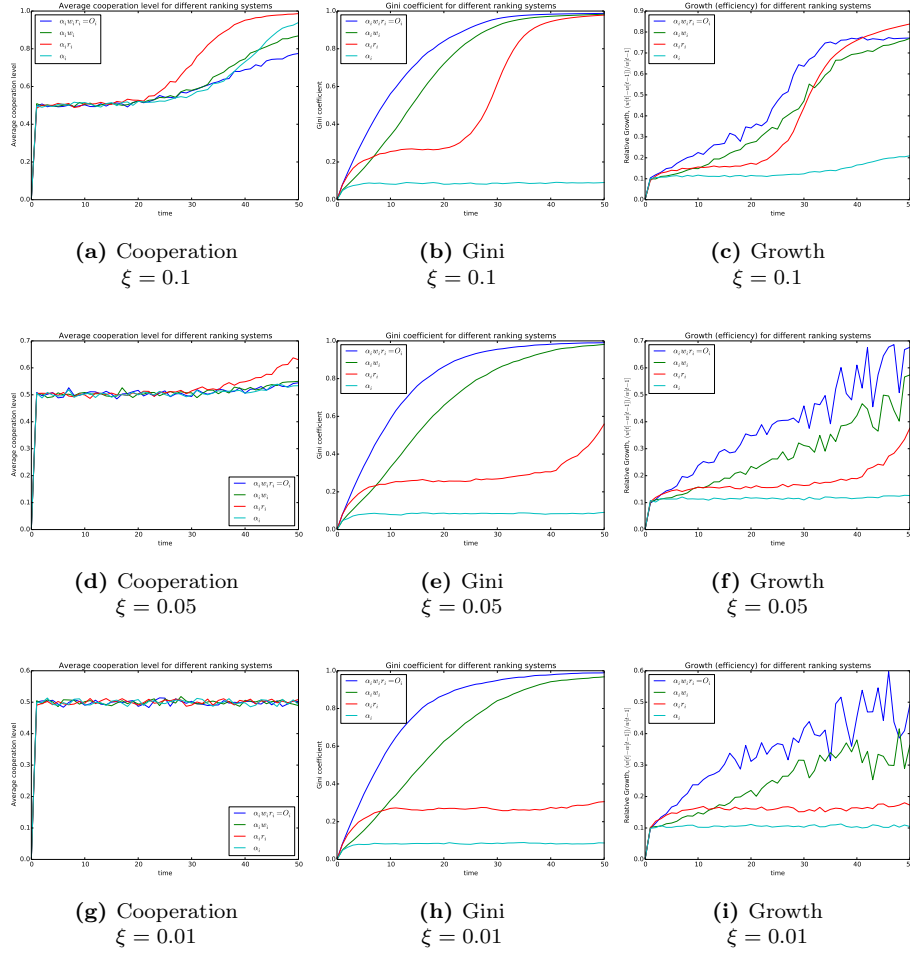


Figure 1: New learning schema results with $N = 1200$ players for $\xi = \{0.1, 0.05, 0.01\}$ during 50 iterations.

2 Equal talent comparison

I ran simulations with equal talent 1 for each player for all three schemas: the new (NLS) and the old (OLS) learning and the Nash equilibrium. The simulations had $N = 500$ players and lasted for $T = 50$ time steps.

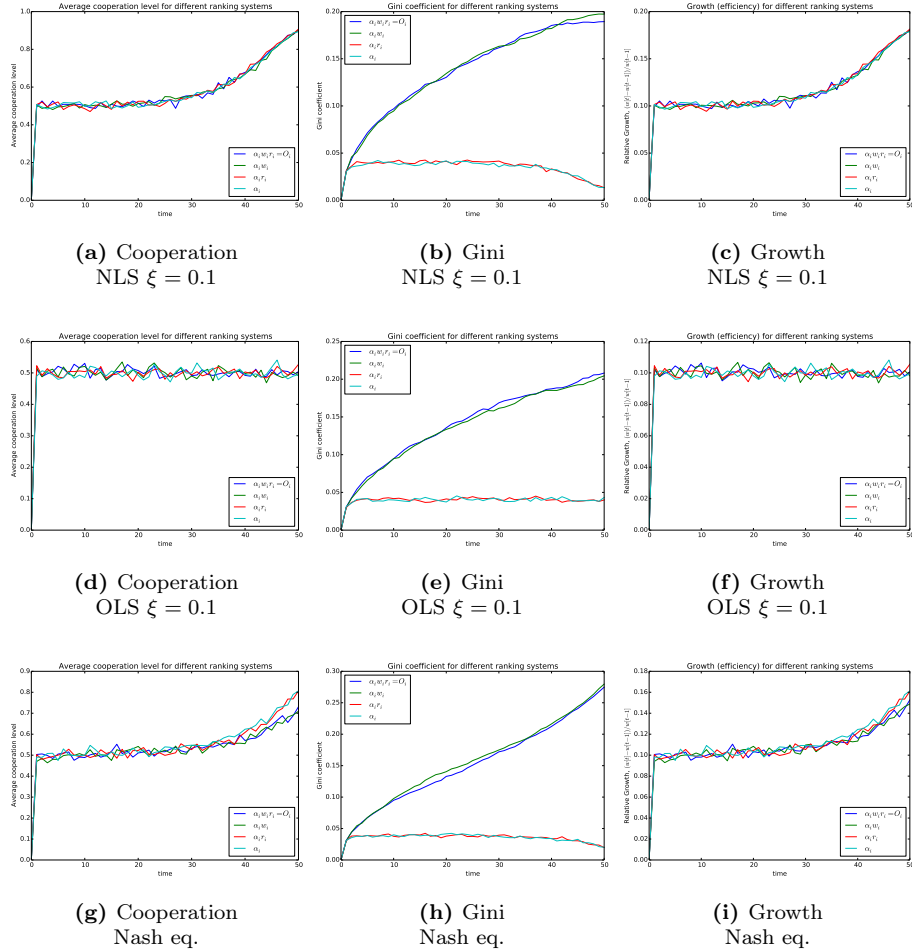


Figure 2: All agents have talent 1. NLS and OLS indicate the new and old learning schema correspondingly. Number of agents $N = 500$, simulation duration $T = 50$.

3 Nash eq blue line

I also had to try and see why did the blue line (grouping according to the output) start falling at some point. It seems to me that for some reason this grouping method for the Nash equilibrium gives a rather jumpy growth and the falling effect was just a statistical feature. On the next page you can see the last twelve time steps for the grouping by output size. On those plots the size of the circles indicate the output size. It can be seen, that the position of the agents jumps around very much, due to which we also see a lot of jumping around for the growth plot in Fig 3 (c).

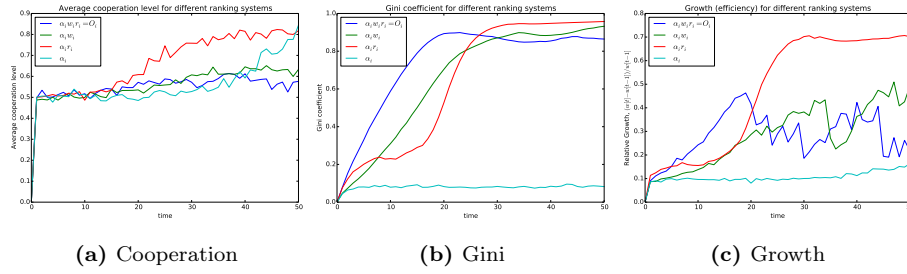


Figure 3: Nash eq. simulation for $N = 500$ during $T = 50$ timesteps.

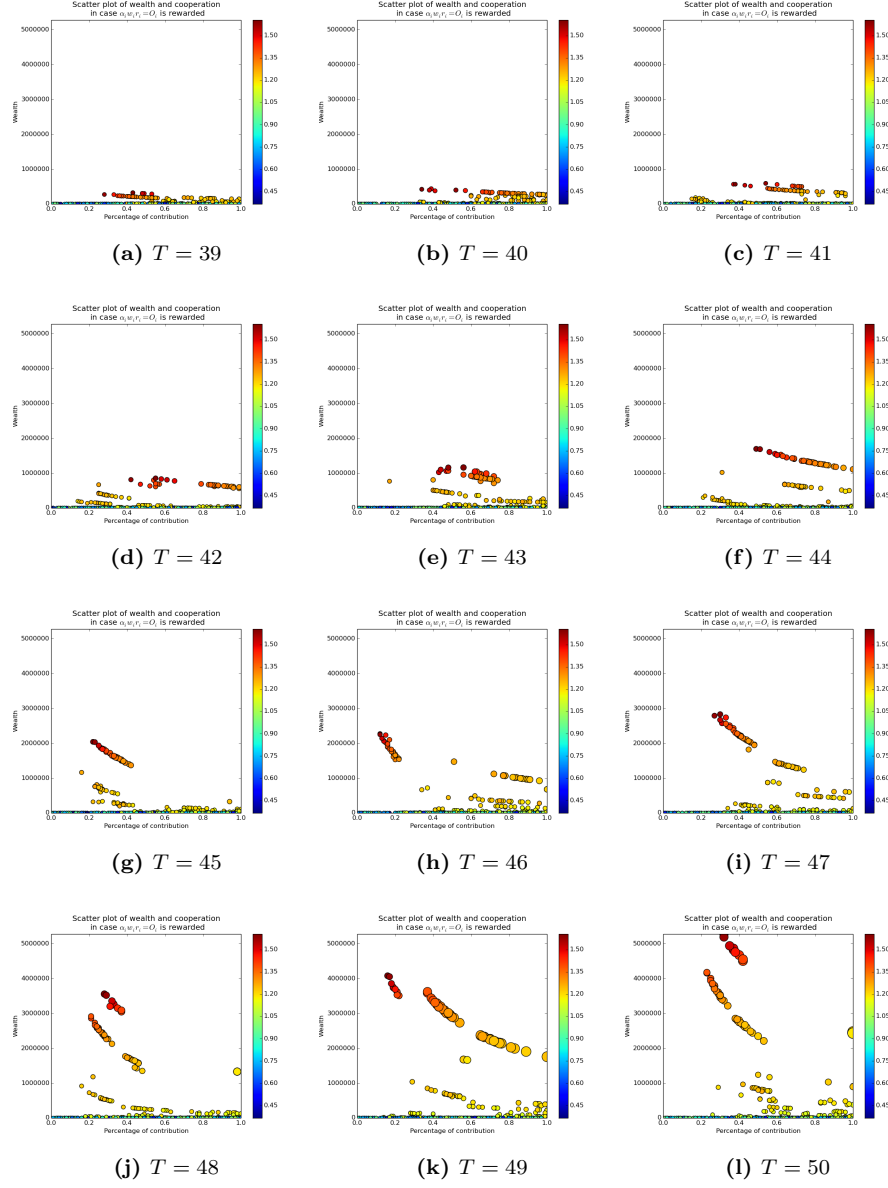


Figure 4: Nash eq. simulation for $N = 500$ with grouping by the output value $O_i = \alpha_i r_i \omega_i$ for the twelve last time steps. The size of the circles indicate the value of the output.