Progress report: Learning vol3
Longer simulations

Madis Ollikainen

July 1, 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 $M_i j$ for the all strategies j. The update $M_{ij}(t) \rightarrow M_{ij}(t+1)$ is done as

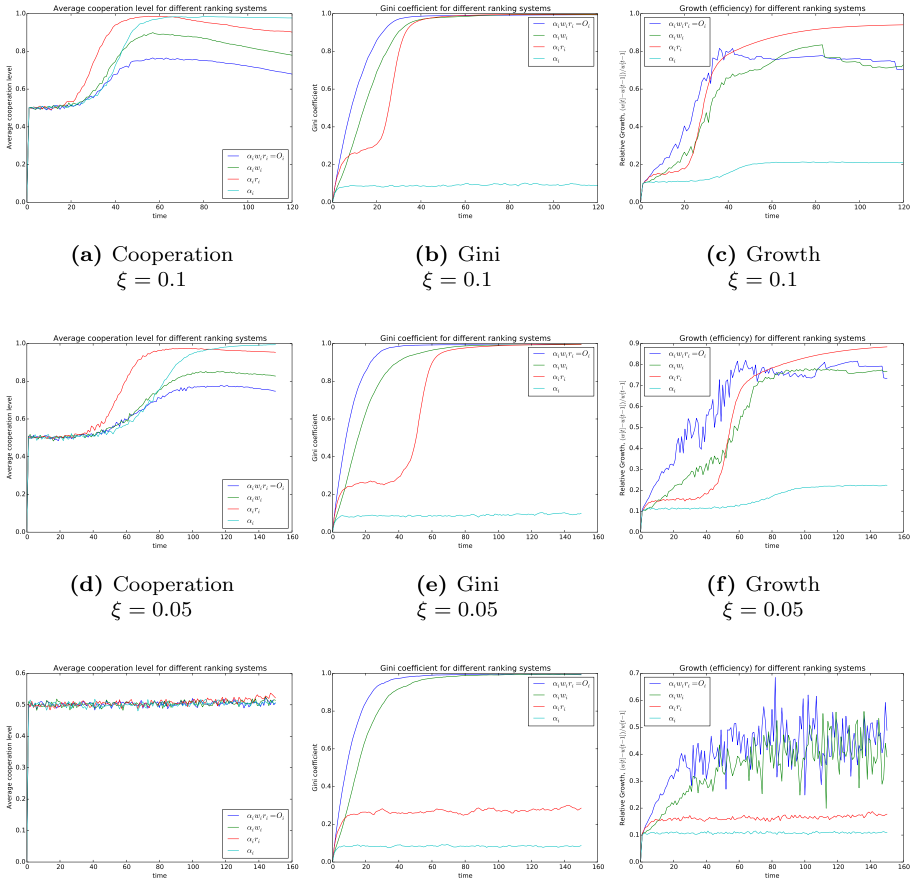
$$M_{ij}(t+1) = M_{ij}(t)\left(1 + \xi \frac{gain_{ij}}{w_i(t-1)}\right), \tag{1}$$

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 $T = 150$ time steps for $\xi = \{0.05, 0.01\}$ and $T = 120$ for $\xi = 0.1$ (the last case went out of bounds for larger time steps) are shown in the graphs below. The case for $xi = \{0.1, 0.05\}$ seems mostly identical, but for the fact that in the case of $\xi = 0.05$ the rise of inequality for the red line (αr ordering schema) is delayed compared to the case of $\xi = 0.1$. For $xi = 0.01$ this rise doesn't occur, but it might well be that it is just delayed a lot more. Also the blue and green lines (ordering schemas depending on the wealth) a lot more spread out and jumpy. This is probably an indication that for the case of $\xi = 0.01$ the learning is much slower and less effective (remember: ξ was the learning factor, see eq. (1)).



(a) Cooperation
$\xi = 0.1$

(b) Gini
$\xi = 0.1$

(c) Growth
$\xi = 0.1$

(d) Cooperation
$\xi = 0.05$

(e) Gini
$\xi = 0.05$

(f) Growth
$\xi = 0.05$

(g) Cooperation
$\xi = 0.01$

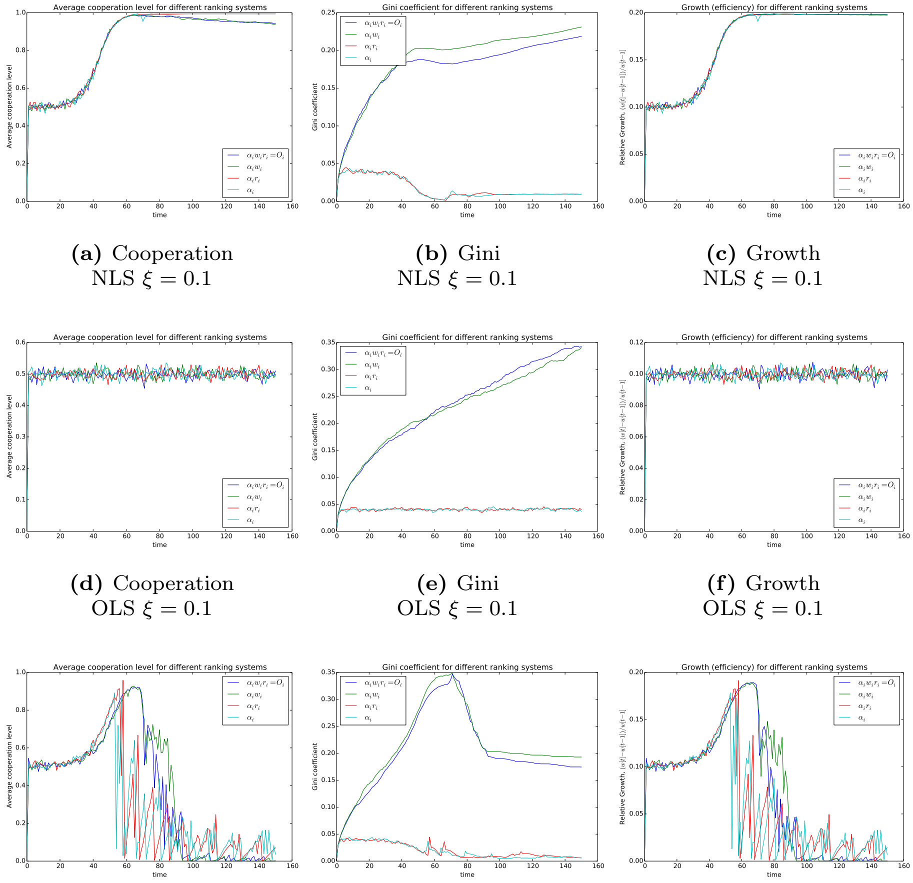
(h) Gini
$\xi = 0.01$

(i) Growth
$\xi = 0.01$

Figure 1: New learning schema results with $N = 1200$ players for $\xi = \{0.1, 0.05, 0.01\}$ during $1T = 50$ iterations for $\xi = \{0.05, 0.01\}$ and $T = 120$ iterations for $\xi = 0.1$.

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 $T = 150$ time steps. The resulting plots (Fig. 2) definitely show that the learning schema has different final results than the Nash eq. simulation. **Stefano, any comments/ideas on this (aka. is this interesting)?**



(a) Cooperation NLS $\xi = 0.1$

(b) Gini NLS $\xi = 0.1$

(c) Growth NLS $\xi = 0.1$

(d) Cooperation OLS $\xi = 0.1$

(e) Gini OLS $\xi = 0.1$

(f) Growth OLS $\xi = 0.1$

(g) Cooperation Nash eq.

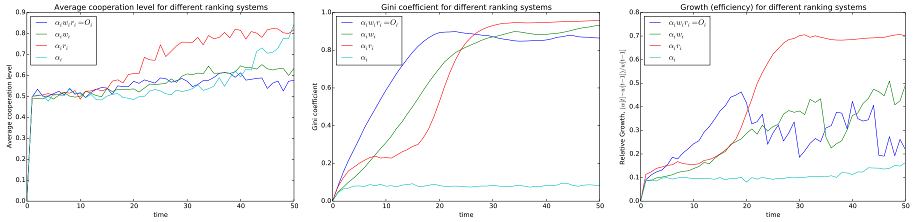
(h) Gini Nash eq.

(i) Growth Nash eq.

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 $T = 150$.

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).

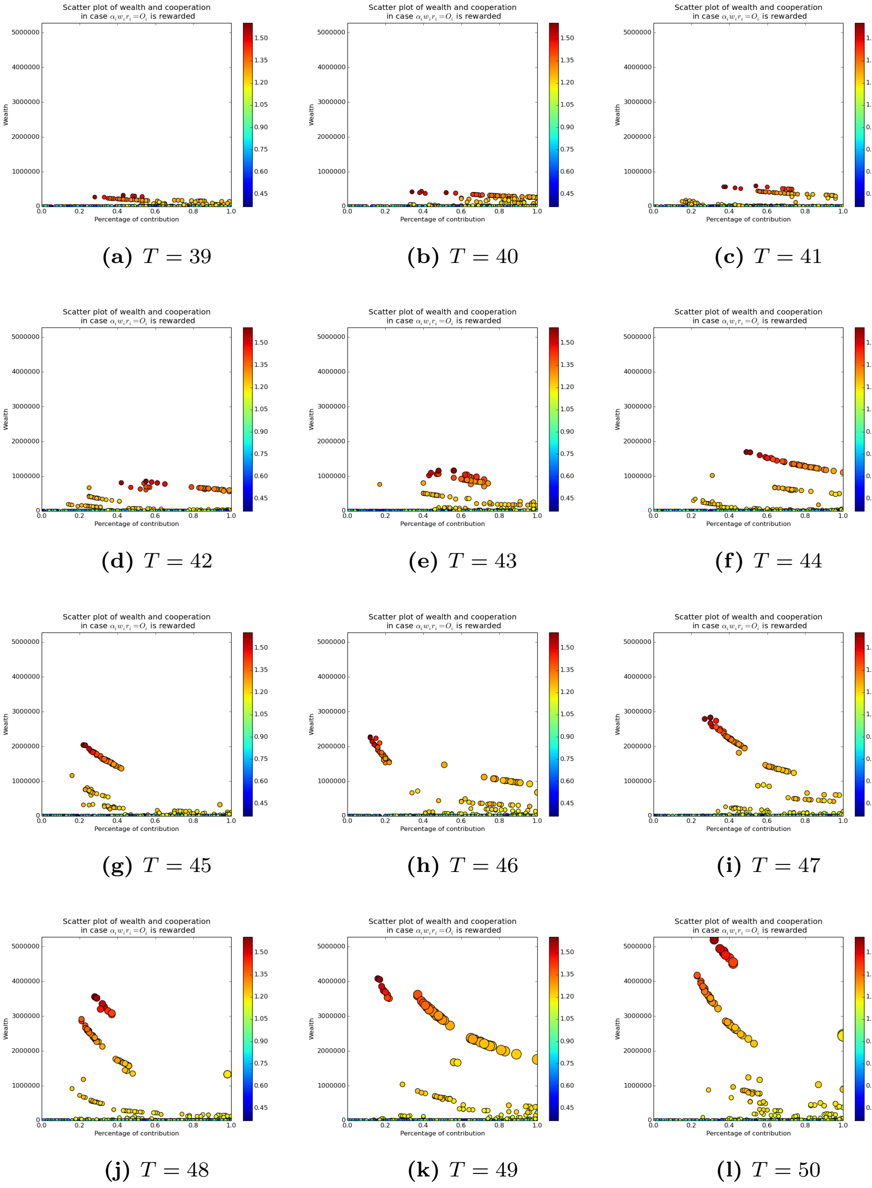


(a) Cooperation

(b) Gini

(c) Growth

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



(a) $T = 39$

(b) $T = 40$

(c) $T = 41$

(d) $T = 42$

(e) $T = 43$

(f) $T = 44$

(g) $T = 45$

(h) $T = 46$

(i) $T = 47$

(j) $T = 48$

(k) $T = 49$

(l) $T = 50$

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.