Complexity Explorer

Spring Challenge 2018

Full text

Complexityexplorer.org/challenges

Consider the following system:

Suppose there are **fifty** (50) agents. At each time step, each agent must decide to locate at one of three possible pools (investing options) described below. These pools are called: **stable**, **high**, and **low**. Agents must choose their pool without knowing what the other agents have picked, and can only rely on information from prior time steps (in particular, each agent must make their choice knowing only the number of agents (but not their identity) that located at each pool and each pool's payoff for all prior time steps). Agents are allowed to switch pools at the start of any time step, but to do so costs the agent a payment of tau (where 0 <= tau). You are not charged tau for your first choice of a pool.

Once all agents have selected a pool, each pool provides a payoff for the time step.

- Any agent that locates at the **stable pool** always receives **$1** at the end of the time step. The other two pools each pay a random amount (see below) that must be split evenly among all of the agents that selected that particular pool at that time step. For example, if twenty (20) agents had chosen a pool that paid $40 that period, each agent would receive a payment of $2 ($40/20 agents) for that period.

The payoffs of the remaining two pools are as follows:

- The **high pool** pays **$80** (that must be split evenly among the residents of that pool) with **probability0.25**and **$0** otherwise (**0.75 probability**), and the ***low* pool** pays **$40** (again, split evenly) with **probability 0.5**and **$0** otherwise (**0.5 probability**). The random payoffs for the high and low pools are independent of one another.

The system runs for 100 time steps, with agents accumulating payoffs at each step.

Your challenge is to:

Explore the general dynamics of the above system, using a diverse set of agents, i.e., agents that have different rules for deciding which pools to invest in.

Analysis of the System

Your analysis should include (but is not limited to):

- What general behaviors arise in this system? How does the wealth of the agents change over time? At the aggregate level? At the individual level?
- How does the diversity of strategies influence the dynamics of the system?
- Are there generally classes of agent behavior (say, based on what data they use, how they process it, or the agent's overall sophistication) that lead to better performance?
- What happens to the system if you violate one of the original assumptions of the problem and allow the agents to alter their strategies over time by observing the performance and strategic details of the other agents?
- Suppose that meta-agents exist that can coordinate the behaviors of a subset of the agents (and split the resulting payoffs equally across the subset)---how does this impact the system's behavior?
- How do the answers to the above questions change as:
    - tau is altered?
    - you change the total number of agents in the world?

Solution guidelines: Follow the questions carefully and be sure to address them all in your analysis. Provide a coherent write up and analysis, no longer than five pages, excluding tables and figures, in PDF format. Create a clear, concise video description of the challenge and solution, no longer than three minutes, and no larger than 25mb, in mp4 format. Develop some general theoretical principles of this system. Suggest related literature and future research directions.

The Tournament

Submission Deadline: June 3, 2018

Please submit your best, single-agent strategy (include both pseudo-code and the actual code) to enter into a tournament based on the original rules. You can assume that 0<=tau<1 and the tournament length will be 100 rounds. There will be 50 agents in each tournament, some of these agents will be other entries, some of these agents may be copies of your own entry, some of these agents will act randomly every round, and some of these agents will choose a pool at the beginning and never move. Depending on the number of entries we receive we may have to have an elimination bracket for the final tournament contest.

Write a NetLogo function of the following form:

to-report choose-strategy-ID

[low-payoff high-payoff low-number high-number my-payoffs my-choices]

```
CODE CODE CODE CODE

report pool

end
```

Where ID is the unique ID number you will be given for this contest, CODE is the code that you have written and pool is an integer where 0 indicates that you want the to be in the stable pool, 1 indicates that you want to be in the low pool and 2 indicates that you want to be in the high pool.

low-payoff and high-payoff are lists where the first element in the list is the payoff received by the low (high) pool in the last round, the second element is the round before that all the way to the beginning of the tournament. Payoffs are reported per agent, but if no agents choose the low pool or the high pool then the value that will be recorded is the same as if exactly one agent had chosen that pool. This is to prevent division by 0 errors, but is easily identified by the fact that the low-number / high-number elements discussed below will be 0 for that time period.

low-number and high-number are lists where the first element in the list is the number of agents who chose the low (high) pool in the last round, the second element is the round before that all the way to the beginning of the tournament.

my-payoffs is a list where the first elements indicates the payoff that your received in the previous round, the second element is the round before that all the way to the beginning of the tournament.

my-choices is a list where the first elements indicates the choice that your agent made in the previous round, the second element is the round before that all the way to the beginning of the tournament.

In addition to the inputs passed in, you can assume that your turtle which is the calling agent has a property called my-data. my-data is an empty list that you can use to store arbitrary data about your agent.

You will also have access to tau which is a global variable that is the current value of tau. tau will be set to a draw from U[0,1) at the beginning of the each tournament.

In addition, to the variables and properties described above, please do not use any of the following variable names in your function since they are used by the simulation code: entry?, wealth, my-pool, myp myc, my-strategy, lowp, highp, lown, highn, current-lowpay, current-highpay, current-lown, current-highn.

You will initially be placed in a random pool, and if you move in the first tick of the tournament to a different pool you will not be charged tau. After the first tick, every time your agent switches pools you will be charged tau. The winner of the tournament will be the agent which accumulates the most wealth after all rounds have finished.