PARETO

Undergraduate Journal of New Economists

University of Toronto Mississauga

Title:	Modeling Social Learning Using Dyna-Q and Ant Colony Optimization
Author:	Renato Zimmermann
Source:	*PARETO: Undergraduate Journal of New Economists*, Vol. 1, Winter 2023
Published By:	Department of Economics, University of Toronto Mississauga
Stable URL:	https://www.uecutm.ca/pareto-zimmermann2023

Economics

UNIVERSITY OF TORONTO

MISSISSAUGA

Modeling Social Learning Using Dyna-Q and Ant Colony Optimization

By Renato Zimmermann

This paper introduces a novel way of modeling social learning in macroeconomics using techniques from model-based reinforcement learning and ant colony optimization. The work extends previous works in bounded rationality and social learning by providing tools to complement previously-distinct models in adaptive learning. We test these new techniques using simulations of job search and consumption. Results demonstrate that models fit using the proposed techniques can learn core economic behaviours given no information about the environment, but do not fully fit reward functions in line with rational expectations theory.

Keywords: *Optimization, Machine Learning, Dyna, Learning Algorithms, Adaptive Learning, Job Search, Information Economics*

I. Introduction

The way humans perceive their environment dictates how they make decisions; this perception is ever-changing, based on previous perceptions and influenced by personal experience. Transitioning from one belief to another is often volatile and subject to the whims of those who influence public information the most. Work in cognitive psychology and behavioural economics have shown that consistent biases in decision-making are possible and that herd behaviour on the part of individuals can lead to irrational decision making [Tversky and Kahneman (1982)] [Lin et al. (2013)]. This paper aims to provide a computational framework on which to model information transmission between agents in social settings using minimal assumptions of agent rationality or knowledge.

The rational expectations (RE) framework dominated macroeconomic models during late twentieth century, and to a large extent remains the predominant technique used in modern models [Muth (1961)]. The theory came as an improvement over earlier work using adaptive expectations, which modeled agents' forecasting as a weighted moving average over past values of the predicted variable [Fisher (1911)]. Unlike adaptive expectations, rational expectations theory treats agents' aggregate predictions to be as good as the actual expected value of what they are trying to predict plus some idiosyncratic error. In light of the empirical results that brought RE models to prominence, the field of bounded rationality has aimed to reconcile these models with the possibility of irrational decision making amongst agents. Adaptive learning (AL) is a popular approach among these models, treating agents as non-perfect — but possibly unbiased — forecasters [Evans and McGough (2020)].

This paper aims to extend the branch of AL known as *social learning*, first developed by Arifovic Arifovic (1994). Social learning aims to model how one agent learns in the context of other agents' decisions as well as their own. We contribute to the literature by using the Dyna-Q model-based architecture [Sutton (1991)]. In this respect we represent societal beliefs with a distributed fitting approach inspired by Ant Colony Optimization [Dorigo et al. (2006)]. The techniques proposed here also enable models that connect both social and individual learning, unlike previous techniques that relied on one approach.

We also explore how individual learning can take place without full knowledge of the environment's dynamics and rewards through approximated dynamic programming. Although exact dynamic programming methods yield quick and unbiased solutions, the procedure's transition from

one state of belief to another is not informative in and of itself. The methods outlined in this paper maintain key properties of commonly used dynamic programming techniques while introducing decentralized information dynamics based on socially-learned models and possible channels for misinformation. This work will enable a more realistic study of how autonomous agents influence the dissemination of information based on experience in an environment with unknown dynamics. Overall this paper aims to provide an intuitive framework to model both social and individual learning in tandem, while keeping assumptions about agents' knowledge of underlying stochastic processes to a minimum. This paper is accompanied by an open-source library capable of replicating the results presented here and written with the objective of being easily extended. This framework can serve to facilitate further exploration of social learning and agent-based modeling in economic research.

II. Related Work & Motivation

A. Economic Modelling

One of the first formulations of agent-level forecasting in macroeconomic models is attributed to the work of Irving Fisher in the form of adaptive expectations [Fisher (1911)]. A simple formulation treats agents' forecasting as an integrated moving average IMA(1,1) prediction. Value expectations are simply the previous forecast corrected by its error weighted by some constant coefficient. Using the Cobweb model as a simple example, agents would forecast prices to be:

$$p_t^e = p_{t-1}^e + \lambda(p_t - p_{t-1}^e) \tag{1}$$

where we denote p_t^e, as the expected price forecast at time t. This is solved given p_t, the actual price at time t, and λ a constant coefficient. This simple formulation served as the backbone to a large body of work in the early twentieth century, yet representing forecasting in such a simple manner allowed for counterintuitive behaviour on the part of agents. For example, as the "error correction" term in equation 1 is constant, agents could consistently underestimate prices that trend upwards. This goes against the intuition that agents are rational in nature, and capable of comprehending simple trends.

The rational expectations hypothesis was first posed by John Muth in an effort to reconcile the notion that agents are, in aggregate, rational in nature and capable of forward-looking forecasting [Muth (1961)]. Instead of forecasting values solely using past observations, as was done in adaptive expectations, agents in RE models take into account all available information about a random variable. Their forecasts are the actual expected value of said variable and only wrong by random idiosyncratic noise[1]. Continuing our example using the Cobweb model, price expectations for a representative agent in the rational expectations model would become:

$$p_t^e = \mathbb{E}_{t-1}[p_t] \tag{2}$$

where $\mathbb{E}_{t-1}[p_t]$ denotes the expected value of price at period t given information available at period $t-1$. This formulation allows for a forward, rather than backward, view of agent predictions.

Work on rational expectations was further developed and popularized by Lucas and now serves as a basis for modelling decision making in macroeconomic models [Lucas (1981)]. Regardless, this

[1]It is important to emphasize that the model does not assume that *individual* agents have this knowledge, but rather that their aggregate predictions would be the same as that of a representative agent fully aware of the distribution and parameters of a random process

framework might not fully conform to the realities of how decisions are made in various situations. The assumption that aggregate forward-looking decisions are equal to the true expected future value is arguably too confident of the aggregate capability of agents, as it would imply knowledge of exact distributions and parameters. Granted this argument is sound when dealing with variables of broad interest, it is less convincing for situations with limited price feedback, unsophisticated agents or information frictions.

In order to tackle this issue, the study of bounded rationality has shown that it is possible to limit the capacity of agents' predictions while preserving key theoretical results from rational expectations theory. The adaptive learning (AL) approach has grown to prominence modeling boundedly-rational agents. The approach is based on the *cognitive consistency principle*, which states that agents are just as good at forecasting as economists [Evans and McGough (2020)]. Instead of setting agent predictions to a value's conditional expectation as in RE, Adaptive Learning allows for misspecification of models and other scenarios on the path towards equilibrium. Consider the common implementation of the AL framework known as least-squares learning. As the name suggests, predictions are made through a least-squares estimate on observable variables. The price prediction from before becomes:

$$p_t^e = \alpha'_{t-1} + \beta'_{t-1} w_{t-1} \tag{3}$$

where w_{t-1} is a vector of observable variables in period $t-1$ and α' and β' are least-squares estimates of the perceived model:

$$p_t = \alpha + \beta w_{t-1} + \varepsilon$$

Predictions done this way bridge the backward-looking view from adaptive expectations while letting agents act rationally, yet bounded in knowledge. Another implementation of AL formulated by Jasmina Arifovic uses the concept of social learning to guide how agents make decisions [Arifovic (1994)]. The original social learning technique consists of using a genetic algorithm (GA) to model the evolution of agent's decision making through adaptation and market selection pressures. The algorithm is structured to resemble the process of genetic mutation and evolution that happens in the natural world. One key assumption backing the use of GA is that analogous processes happen in economic environments through a market selection process. The GA approach is unlike those previously outlined in this section, as it specifies agent decisions directly, instead of deriving them as a function taking in some expected value as an input. In the Cobweb model example used thus far, while previous approaches would first generate a price prediction used as an input to a decision function, the GA approach produces this decision directly by encoding the agent's genes. Consider a production function as an example. While the previous approaches would use expected price p_t^e as the input to a production function the GA approach would model the production decision directly as in equation 4:

$$q_{i,t} = \frac{1}{\bar{K}} \sum_{k=1} a_{i,t}^k 2^{k-1} \tag{4}$$

where $q_{i,t}$ is agent i's production at time t and $\bar{K}$ is a normalizing parameter. Genes in the original social learning paper are strictly binary; in equation 4, firm i's k^{th} gene at time t is represented by $a_{i,t}^k$. This approach treats decision-making as an evolving problem, where genes that inform decisions are selected based on which combination yields the best production strategy over time.

Using a genetic algorithm as presented above still lacks a detailed treatment of how agents could use other information available to them, or how actual learning can take place beyond the agent's "genes".

Unlike previous work in adaptive learning, Arifovic's social learning model approaches the problem through an agent based modeling (ABM) standpoint. Developments applying ABM in the context of macroeconomics have been sparse [Farmer and Foley (2009)]. Nonetheless, ABM applications in game theory have seen significant strides in the past few years due to an increased interest in multi-agent reinforcement learning. Research using GA to inform policy decisions has continued recently in the context monetary economics [Arifovic et al. (2020)]. One notable example applying ABM (not GA) to policy-making was developed by Zheng et al. [Zheng et al. (2020)]. The work formulates the issue of optimal taxation as an OpenAI-Gym-style problem and uses deep reinforcement learning to model both decision makers and policy makers [Brockman et al. (2016)].

Taking inspiration from previous approaches to social learning and the recent advancements in reinforcement learning, this paper aims to outline a framework on which social learning can take place in tandem with individual learning. This comes as an effort to supplement the lack of a social aspect to learning in the AL literature while also allowing for a more detailed treatment of the learning process that happens in economic environments. In order to conform with the intuition that agent's rationality is bounded by the information they have been given, a fitting procedure is outlined using adaptive dynamic programming techniques that allow the study of learning behaviours when agents start with no knowledge of environment dynamics or rewards.

B. Learning Modelling

Ant Colony Optimization (ACO) is defined as a metaheurisitc general workflow that can be modified to solve several optimization problems. Drawing inspiration from the behaviour of wild ants, the workflow is characterized by autonomous agents that traverse the state space leaving behind pheromones. Pheromones can in turn influence the decision of future agents. Pheromones mix with known transition costs to guide agents to a decision; these pheromones aim to incorporate future costs, such that prospective ants avoid short term decision-making.

This class of swarm intelligence algorithm was first used to approximate a solution to the Traveling Salesman Problem (TSP), where the objective is to find the shortest path on which all nodes in a given graph are visited. ACO has previously been used in place of dynamic programming to approximate a solution to the problem, as it offers better scaling as state spaces grow. As such, it has become a common alternative to solve $\mathcal{NP}$-hard problems.

However, economic applications do not conform to several aspects of the vanilla TSP that motivated the original algorithm. The most prevalent difference lies in how the economic problems we are trying to study are stochastic processes; whereas state-transition costs are fixed in the TSP. Still, the metaheuristic has been used in the context of economics for portfolio optimization, option pricing and job shop scheduling [Hsu (2014)] [Deng and Lin (2010)] [Kumar et al. (2009)] [Huang and Liao (2008)]. This work also takes inspiration from earlier work done using ACO to find stochastic shortest paths and approximate solutions to the dynamic traveling salesman problem [Horoba and Sudholt (2010)] [Guntsch et al. (2001)]. Although unrelated to economics, these problems are comparable to dynamic inter-temporal decision-making in macroeconomic models.

The general ACO workflow is encompassed in Algorithm 1 followed by a discussion of formulas and parameters common to most uses of the metaheuristic. We outline modifications made to this framework in Section III.

Algorithm 1: Ant Colony Optimization Metaheuristic

```
initialization
while termination condition is not met do
    for each ant generated do
        construct solution
    for each path history and solution do
        update pheromones
    pheromone evaporation
```

This framework offers great flexibility in how certain steps are implemented, such that it can be modified to approach numerous problems[2].

In the original Ant System (AS) formulation by Dorigo Dorigo et al. (2006), heuristic information is weighted with pheromones to form transition probabilities, which can be used as stochastic weights in a decision making step. This weighting is shown is equation 5.

$$p(s,a) = \frac{\tau(s,a)^{\alpha} \cdot \eta(s,a)^{\beta}}{\sum_{a' \in A(s)} \tau(s,a')^{\alpha} \cdot \eta(s,a')^{\beta}} \tag{5}$$

where $A(s)$ is the actions available at state s and η are the known immediate rewards from taking an action at a given state.

The parameters α, β and ν are commonly preserved independent of the problem at hand; their original usage is shown in equation 5. The α parameter dictates the degree of importance assigned to pheromones; similarly, β controls the degree of importance assigned to known transition costs (broadly referred to as heuristic information). Finally, ν — called the evaporation rate — defines the rate on which past pheromones fade as time passes, making way for newer information. The last parameter is a key part of updating pheromones, which is done as follows:

$$\tau'(s,a) = (1-\nu) \cdot \tau(s,a) + \sum_{k=1}^{m} \Delta\tau(s,a) \tag{6}$$

where $\tau(s,a)$ is the current pheromone for state-action pair (s,a) and $\tau'(s,a)$ is its update; m is the number of ants on which the state-action pair applies to; and $\Delta\tau(s,a)$ is the pheromone left by ant k.

In the context of economic decision making and information dynamics, α and β can be interpreted as the influence decision-making agents give to socially-transmitted information versus long-running beliefs. Likewise, the evaporation rate ν can be seen as the degree on which socially-transmitted information becomes less prominent with time, or the rate on which this information evolves into long-standing beliefs.

[2]In the following equations, we slightly deviate from original notation in order to preserve compatibility with the problem discussed in this paper. Notably, node-to-edge subscript notation (τ_{ij}) to state-action notation ($\tau(s,a)$), where states are interpreted as nodes and actions as edges.

III. Model

A. Dyna-Q

While ACO can model social logic there must also be a mechanism to fit more dynamic models. One suitable choice is the Dyna-Q architecture, due to its compatibility with ACO and its potential economic intuition [Sutton (1991)]. A reasonable choice is a tabular environment with infinite or finite episodes[3].

Algorithm 2: Basic Dyna-Q Algorithm

Initialize $Q : S \times A \mapsto \mathbb{R}$, $M : S \times A \mapsto S \times \mathbb{R}$
while *Condition not met* **do**
  Sample trajectories $\tau_1, \tau_2, \dots \tau_n$ from the environment using policy π.
  Fit $M(S, A)$ using trajectories $\tau_1, \tau_2 \dots \tau_n$.
  for *Planning steps* **do**
    Sample S_{train}, A_{train} randomly from within $\tau_1 \dots \tau_n$
    $R_{train}, S'_{train} \sim M(S_{train}, A_{train})$
    $Q(S_{train}, A_{train}) \leftarrow$
     $Q(S_{train}, A_{train}) + \alpha \left(R_{train} + \gamma \max_A Q(S'_{train}, A) - Q(S_{train}, A_{train})\right)$

This paper implements the Dyna-Q+ algorithm in order to add a time-incentive to explore. This algorithm builds on what we have seen by adding a time-weight parameter κ that artificially increases rewards for older or unseen state-action pairs. The algorithm is the same as Algorithm 2 with two modifications: a) additional variables keep track of when a state-action pair was last seen and b) the following is added to R_{train}

$$R_{train} = R_{train} + \kappa \cdot \sqrt{\Delta t}$$

where Δt is the time since the last time this state-action pair was seen.

The economic intuition behind the choice for the Dyna algorithm lies in how new information is "pooled" into shared models of the economy, which is then used to inform agents of the market landscape. Additionally, the information is integrated into public perception based on trajectories that are not necessarily complete. A real world example of such a model is a newspaper, which surveys common knowledge and past experience to inform how economic agents go on to perceive value.

The Dyna-Q framework also allows us to separate social from individual learning unlike previous macroeconomic models. By keeping the model M and rewards Q separate, we essentially modularise the social and individual learning components of our model. While socially-shared experiences update the shared model of the economy, samples from this model can be used to train individual models, possibly with supplemented private data. Although Algorithm 2 and simulations in Section IV use Q-tables as individual models, it would be possible to use recursive least squares or other techniques commonly used in the AL literature.

B. The Algorithm

Our main algorithm is composed of iterations of generation, decision and updating, outlined in more detail in Algorithm 3. Generation is the process of generating new, finitely-lived agents.

[3]Using a different model-based approach such as MBPO Janner et al. (2019) could allow us to optimize on a continuous environment. The discussion and implementation of this method are beyond the scope of this paper, but likely a subject of future research.

Decisions are made by each agent based on Q values. Each agent will experience the environment differently and keep a history of the rewards earned in their lifetime. The joint history of all agents in a period will be fed to model M, which will in turn influence decisions made by future agents. Other mechanics are based on the Dyna-Q+ algorithm, as we can see in algorithm 3.

Algorithm 3: Main Information Transmission Routine

Initial $Q : S \times A \mapsto \mathbb{R}$, $M^{pher}, M^{belf} : S \times A \mapsto S \times \mathbb{R}$, Time Weight κ
while *Condition not met* **do**
  for *Number of ants* **do**
    Sample trajectories $\tau_1, \tau_2, \ldots \tau_n$ from the environment using policy π.
    Fit $M^{pher}(S, A)$ using trajectories $\tau_1, \tau_2 \ldots \tau_n$.
  for *Planning steps* **do**
    Sample S_{train}, A_{train} randomly from within $\tau_1 \ldots \tau_n$
    $R_{train}, S'_{train} \sim M^{join}(S_{train}, A_{train})$
    $Q(S_{train}, A_{train}) \leftarrow$
     $Q(S_{train}, A_{train}) + \alpha \left(R_{train} + \gamma \max_A Q(S'_{train}, A) - Q(S_{train}, A_{train})\right)$
  Evaporation

A general visualization of the model is presented in Figure 1:

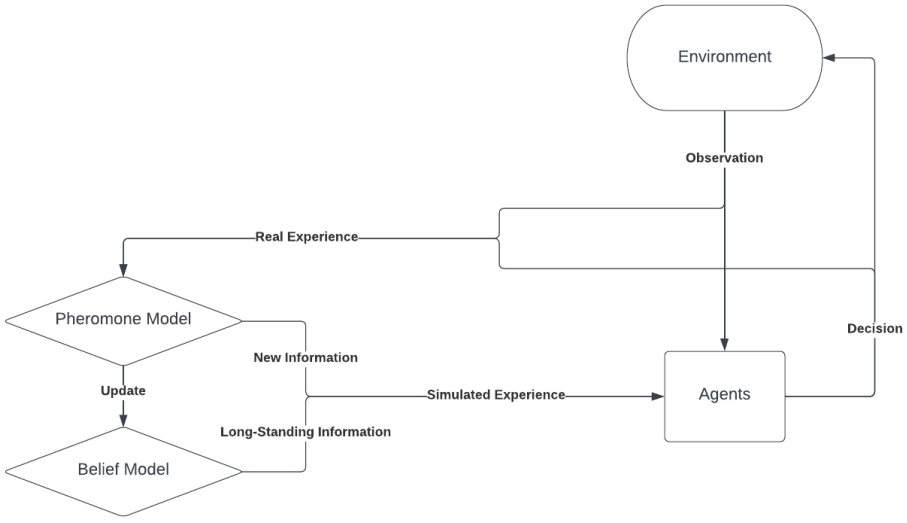

Figure 1. : Model Diagram. The figure shows the basic flow of information through the proposed model. Information begins as observations that are acted upon by an agent and used as training data for a pheromone model. Agent actions set a prior for upcoming observations from the environment. Pheromone models update belief models through evaporation and agents learn from simulated experiences sampled from both of them. Learning will in turn affect how agents make decisions in the future.

C. Generation

Autonomous agents are at the core of the mechanics outlined in this paper, as they allow for a study on the gradual change of beliefs in an environment. Agents need not be homogeneous, however, and can employ any mixture of parameters in order to model a scenario; as outlined in Section II.B, said parameters control agents' perception of the environment and influence they hold on it. Results from Section IV explore a fairly limited scope of parameters, which warrants future work exploring the use of stochastic agent generation and model-mixing in describing learning dynamics.

D. Sampling

Drawing from the ACO metaheuristic, the model in this work separates beliefs from pheromones. Beliefs are interpreted as long-standing expectations of heuristic information and used in conjunction with pheromones in order to form a decision. Model predictions are based on a combination of beliefs and pheromones. This combination is done for both rewards and transition dynamics, which together yield the sample from M^{join}. Here, we separate the reward and transition dynamics pheromones and beliefs, which we denote as R^{pher}, R^{belf}, D^{pher}, D^{belf} respectively. Notice that although we maintain the exponential weighting of dynamics presented in Section II.B, we use a weighted sum of rewards in order to maintain reward predictions similar to those seen in the environment. State-action rewards and dynamic predictions are defined by:

$$R_{train}(s,a) = \frac{\alpha \cdot R^{pher}(s,a) + \beta \cdot R^{belf}(s,a)}{\alpha + \beta} \tag{7}$$

$$p(S'_{train}|s,a) = \frac{D^{pher}(S'_{train}|s,a)^{\alpha} \cdot D^{belf}(S'_{train}|s,a)^{\beta}}{\sum_{s' \in S'(s)} D^{pher}(s'|s,a)^{\alpha} \cdot D^{belf}(s'|s,a)^{\beta}} \tag{8}$$

Models used in this paper employ an ε-greedy deterministic decision procedure, where actions with highest expected rewards are selected $\varepsilon\%$ of the time and randomly elsewhere. Decision are made as follows:

$$a_{t+1} = \begin{cases} \arg\max_a Q(s_t,a) & \epsilon < \varepsilon \\ P(a) = Q(s_t,a) & \text{elsewhere} \end{cases} \tag{9}$$

where $\epsilon \sim U(0,1)$

E. Updating

Updating is the process of changing beliefs on the expected state-action rewards in the current environment based on the agent's lifetime experience. This stage is split into two further parts: fitting and evaporation.

Fitting

The model fitting procedure is based on producing standard Monte-Carlo estimates for rewards and dynamics. One twist is that if we have not seen a certain state-action pair, its transition

probabilities are set to be uniform, as to avoid errors when combining pheromones with beliefs.

Evaporation

Again borrowing from the ACO metaheuristic, we use the concept of evaporation to bring pheromones closer to established beliefs. In addition to that, we use evaporation to allow established beliefs to incorporate newly-acquired information. The evaporation stage is defined by the following two operations:

$$R^{pher} = \begin{cases} R^{pher} & \text{if visited} \\ \kappa & \text{otherwise} \end{cases} \tag{10}$$

$$M^{belf} = (1-\nu) \cdot M^{belf} + \nu \cdot M^{pher} \tag{11}$$

This mechanism allows for general beliefs to converge to long-standing pheromone feedback, such that a "new normal" can be established after a certain amount of iterations following a change in dynamics.

IV. Simulations

A. Experimental Design

In order to test the proposed procedure, we implement environments to simulate the dynamics of known economic models. As the simulations proposed here are tied to certain economic models, the simulations' expressiveness might be limited by what that model aims to describe. That is, although environments used in ABM typically include specially-tailored dynamics not necessarily linked to standard economic models, we aim to remain faithful to our environments' original descriptions.

We propose the implementation of the McCall model of unemployment and the Huggett model of consumption following an OpenAI-Gym-style environment similar to the approach by Zheng et al. [Zheng et al. (2020)] [McCall (1970)][Huggett (1993)][Brockman et al. (2016)].

In the McCall model, agents receive a wage proposal at each step and can either accept or reject it. By default, accepting a wage terminates the episode and gives rewards equal to said wage up to the agent's age of death. The agent receives unemployment benefits if they are not employed in a certain episode, and dies at some pre-set age. This can be extended to let agents quit their jobs or have a chance of being fired at each episode. More formally, McCall agents face the following problem:

$$\begin{aligned} &\max_{t'} \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t Y_t\right] \\ &s.t. \\ &Y_t = b,\ t < t' \text{ and } Y_t = w_{t'},\ t \geq t' \end{aligned}$$

where β is the discount factor and Y_t is the income at time t. This income is equal to some constant unemployment benefit b or a wage $w_{t'}$ accepted at time time t'. The wages follow a beta-binomial distribution, that is, $w \sim BetaBin(n, \alpha, \beta)$.

Similarly, in the Huggett model, agents receive some stochastically-generated income at each step and can use it together with their existing assets to consume goods and gain utility from doing so[4]. Leftover assets or income are compounded by some interest rate and become the assets available in the next period. The agent can also borrow money up to a certain amount, so assets can also be negative. Interest is accrued on debt, which simply increases the amount of debt the agent has. The problem is formulated as the consumer problem:

$$\max_{c_t} \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t u(c_t)\right]$$
$$s.t.$$
$$a_{t+1} + c_t \leq Y_t + (1+r)a_t \text{ and } a_t \geq -B$$

where, c_t is consumption at time t, a_t are assets at time t, r is a constant interest rate, B is a constant borrowing constraint and $Y_t \sim BetaBin(n, \alpha, \beta)$ is an income shock at time t.

For each of the environments, we test four separate algorithms: Q-Learning, DynaQ, DynaQ+ and ACO DynaQ+.

We aim to compare how these perform in terms of speed, fitting patterns, optimal Q-values and average lifetime utility with greedy policies. It is important to mention that environment-specific intuitions shall be applied to some of these analyses. Firstly, optimal Q-values should reflect some sort of economic conclusion these models aim to answer. In the case of the McCall model, the moment where accepting a wage (action 0) brings higher expected utility than rejecting it (action 1) is called the *reservation wage* and should happen only once. In the case of the Huggett model, we should observe that higher asset levels warrant higher spending, and that lower asset levels risk bankruptcy, thus calling for reduced spending. Secondly, a sanity test for any of our algorithms in the Huggett model is whether an agent goes bankrupt, that is, gets -2000 utility at some point in the test. This has obvious signs in the results, and shouldn't happen in a well-fit algorithm. Note that this might still happen often during training, as bankruptcy is achievable at any state and not much exploration is needed to get there. The appendix contains a complete list of initialization parameters for each object used in the experiment.

B. Results

We begin by considering the comparaive speeds between the algorithms:

	Mean Rewards		Fitting Time	
Environment	Huggett	McCall	Huggett	McCall
Optimizer	(Utility)	(Income)		
Q-Learning	-1.698124	410.865750	84.621856	81.025059
DynaQ	-1.695528	417.330448	480.070568	470.455797
DynaQ+	-1.695875	422.447582	506.255441	494.747565
ACO DynaQ+	-1.756124	405.878000	92.554653	90.506384

Table 1—: Big Run Results. Values show the test-time performance of agents after $100,000$ Q-value updates.

[4]Here, we use a CRRA utility function with parameters described in the Appendix. A small quantity is also added to the denominator such that the punishment for zero consumption is non-infinite

	Mean Rewards		Fitting Time	
Environment Optimizer	Huggett (Utility)	McCall (Income)	Huggett	McCall
Q-Learning	-1.687222	410.362499	7.350350	7.597060
DynaQ	-1.698818	408.935622	44.519349	45.034055
DynaQ+	-1.689297	417.272493	47.351125	46.833290
ACO DynaQ+	-1.888333	402.394940	8.287741	8.021743

Table 2—: Medium Run Results. Values show the test-time performance of agents after $1,000$ Q-value updates.

	Mean Rewards		Fitting Time	
Environment Optimizer	Huggett (Utility)	McCall (Income)	Huggett	McCall
Q-Learning	-1.833195	379.658680	0.094567	0.076132
DynaQ	-1.696222	396.769244	0.460102	0.467019
DynaQ+	-1.720275	390.504100	0.487222	0.493065
ACO DynaQ+	-2.068486	392.217915	0.101478	0.100523

Table 3—: Small Run Results. Values show the test-time performance of agents after 100 Q-value updates.

The tables were gathered from running each algorithm with $100,000$ Q value updates, which we refer to as our "big run". We compare this run with two smaller runs, which we will refer to as our "medium run" ($1,000$ Q updates) and "small run" (100 Q updates). Note the ACO DynaQ+ algorithm has fewer iterations (we divide the number of iterations for the other algorithms by the "Ants" parameter) in order to equate it to other algorithms in terms of Q value updates. As can be seen, the Huggett model has consistently higher fitting times than the McCall model. This is to be expected, as the Huggett model is an infinite-horizon model, while the McCall model terminates when a wage is accepted. Additionally, we can clearly see how the model fitting and sampling come with an extra cost in terms of run speeds, although this cost seems to be decreased in the ACO DynaQ+ algorithm. This is clearly due to the reduced number of iterations used in the algorithm, although it does say something about the approach as an in-between for Q-Learning and Dyna algorithms. The last point is further backed by our performance results, which seems to favour the original Dyna Algorithms. This holds in the big run but favours the ACO Dyna variation when the run is short. Note that regardless Q-Learning still performs best in all runs.

The test performances give a positive picture of our algorithms, although it undermines the need for the Dyna-like algorithms. Our tests are composed of 100 runs of 100 episodes using a purely-greedy function, which always picks the action with highest Q value. Firstly, we notice that given enough time, our algorithms are able to perform better on the models, as seen by the increase in mean rewards over larger runs. Most importantly, all algorithms given a medium or larger number of iterations were able to learn how to avoid bankruptcy and how to better manage one's income. As we shall see, these come at odds with our training curves, although it might just be evidence of the overall volatility of the models and need for more "fragile" exploration. Finally, we note that Q-Learning seems to consistently perform at the same level as the original Dyna variants, with our

ACO DynaQ+ variant lagging behind. This is only different in the McCall model in the small run case, where our variant out-performs other methods.

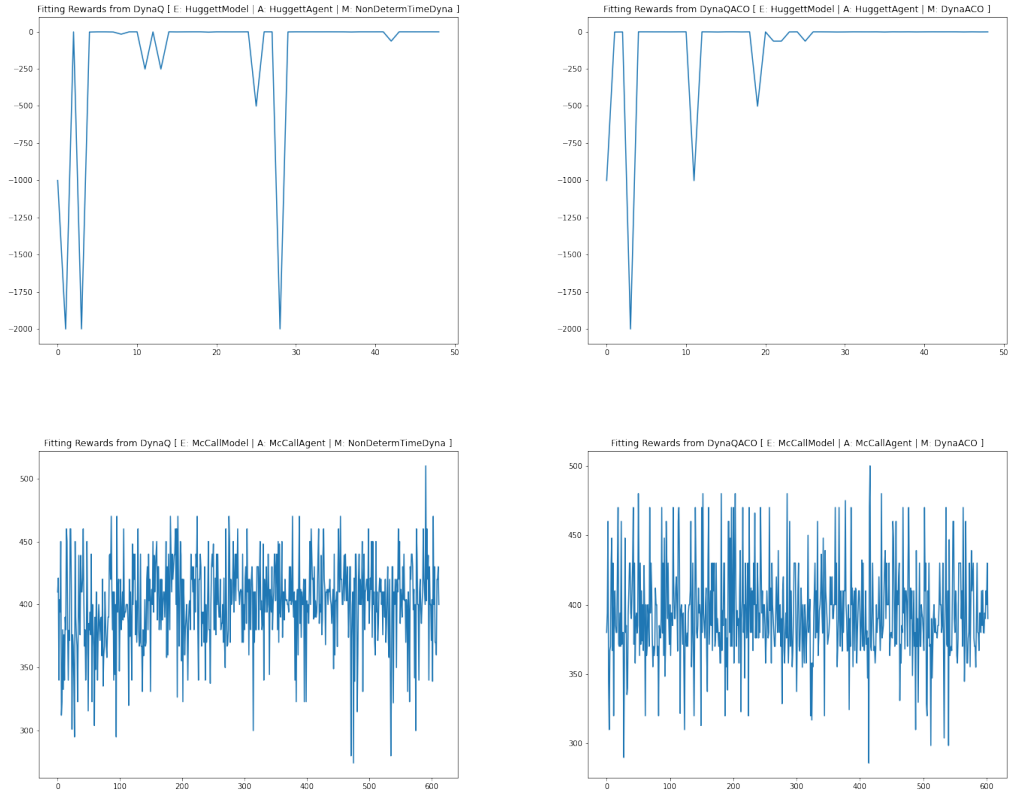

Figure 2. : Selected training curves for medium run. The graphs show the progression of agents' average utility per episode during training. The x-axis represents the episode number and the y-axis represents the average utility achieved in that episode (more is better). Downward spikes in the Huggett model show instances in which agents have no available income to spend in a given period and thus incur a penalty of -2000 utility in that period.

Overall, our fitting patterns are discouraging. As seen the following figures, our medium run fitting curves do not seem to be evidence of much learning. There does seem to be a slightly-upwards trend in some of the fitting curves, although even these do not show clear evidence of learning. Below we display the results of our DynaQ+ ACO model compared to the DynaQ+ algorithm in both environments after a medium run. The noise in the McCall model is to be expected, as state transitions are purely stochastic if a wage is rejected. As for the McCall model, exploration can cause severe dips in the model due to bankruptcy. This causes significant visual deviations in our fitting curves, although the main test should be on how agents act at test-time. Possible explanations aside, these results will put into question the effectiveness of using approaches based on Q-Learning when fitting economic models.

Our Q value interpretation seems to be more promising from the results in the big run, as seen in the figures. Although they are not consistent throughout the algorithms as we would expect from such a significant amount of time to learn the environment. Of the models, the Q-Learning

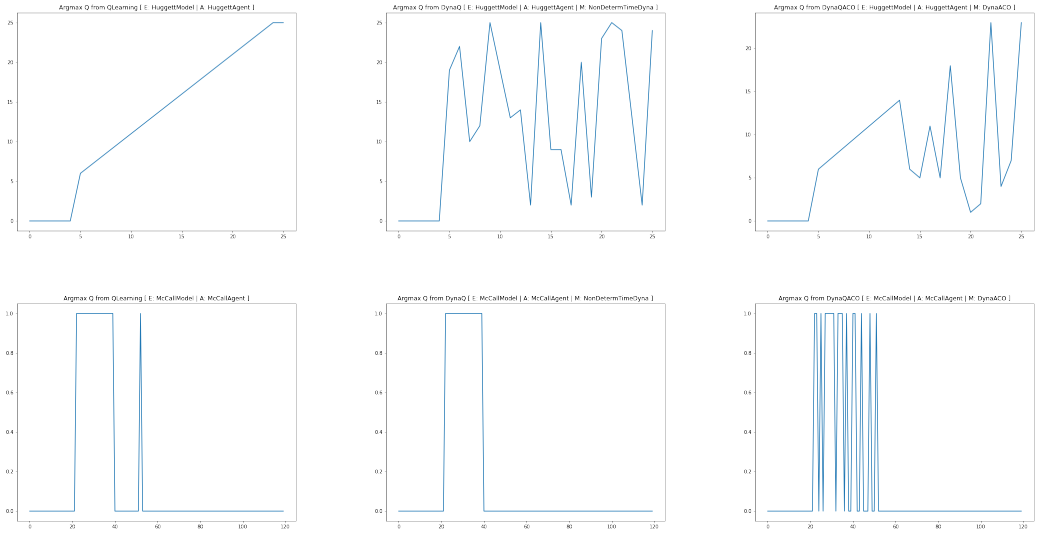

Figure 3. : Argmax over selected Q values following large run. The values represent the optimal actions at a given state as per the Q-table after extensive training. The x-axis shows a continuous state sequence and the y-axis shows the optimal learned action at that state. States in the Huggett model are asset levels and actions are the spending amount. States in the McCall model are wage offers and actions are binary for accepting or rejecting offers.

interpretations are the "cleanest", which suggests that model bias might come at odds with our aim to answer economic questions; thus, we include the Q-Learning results below as well. The McCall model seemed to be the most challenging for algorithms to learn, as is evident from the Q value interpretations. Most algorithms are able to detect the theoretical shift at a proposed wage of 40, although most also have "spikes" elsewhere (except for DynaQ+,). Similar results are present in the Huggett model Q values, although a general upward trend is present in all results; Q-Learning and the DynaQ+ ACO algorithms are able to learn the full curve, which is encouraging. One explanation for the results of our Dyna-like algorithms is model bias and overly-reinforcing a few Q-values rather than exploring. Still, even DynaQ+ did not fully learn the curve, which might be indicative of a low time weight.

V. Conclusions

In this paper, we presented a model of information transmission using model-based reinforcement learning and mechanics from Ant Colony Optimization. We set up the theory to create said model, formulated it as an algorithm, and tested it in the context of two custom economic environments. We have found that although our algorithm satisfies the necessary economic intuition, it does not manage to outperform alternatives in the tested environments. Although our results provide evidence that agents' test-time decision-making conforms to economic intuition, training data shows unsatisfactory increases in utility from learning. In spite of these results, the question remains as to how to most effectively model learning in environments where agents do not know the model dynamics.

In comparison to the standard approach taken by AL and social learning using GA, our algorithms are clearly at a disadvantage. However, further work into the application of the methods outlined here is warranted. Perhaps the models themselves present unrealistic or over-stylized characteristics

that make our approaches especially ineffective. The McCall model, is very close to being a bandit problem, as the next state given a salary rejection is completely random. Similarly, the Huggett model introduces significant stochasticity to our action space, which might likewise imbalance the ease of learning. Perhaps models with increased complexity are necessary to better utilize the algorithms proposed; Although it is possible that the landscape of macroeconomic problems are incompatible with the tools presented in this paper.

REFERENCES

[Arifovic 1994] Arifovic, Jasmina: Genetic algorithm learning and the cobweb model. In: *Journal of Economic Dynamics and Control* 18 (1994), Nr. 1, S. 3–28. – Special Issue on Computer Science and Economics. – ISSN 0165-1889

[Arifovic et al. 2020] Arifovic, Jasmina ; Grimaud, Alex ; Salle, Isabelle ; Vermandel, Gauthier: Social learning and monetary policy at the effective lower bound. (2020)

[Brockman et al. 2016] Brockman, Greg ; Cheung, Vicki ; Pettersson, Ludwig ; Schneider, Jonas ; Schulman, John ; Tang, Jie ; Zaremba, Wojciech: *OpenAI Gym*. 2016

[Deng and Lin 2010] Deng, Guang-Feng ; Lin, Woo-Tsong: Ant Colony Optimization for Markowitz Mean-Variance Portfolio Model. 6466 (2010), 12, S. 238–245. ISBN 978-3-642-17562-6

[Dorigo et al. 2006] Dorigo, Marco ; Birattari, Mauro ; Stutzle, Thomas: Ant colony optimization. In: *IEEE Computational Intelligence Magazine* 1 (2006), Nr. 4, S. 28–39

[Evans and McGough 2020] Evans, George W. ; McGough, Bruce: Adaptive Learning in Macroeconomics. (2020), 11

[Farmer and Foley 2009] Farmer, J. ; Foley, Duncan: The economy needs agent-based modelling. In: *Nature* 460 (2009), 08, Nr. 7256, S. 685686

[Fisher 1911] Fisher, Irving: *The Purchasing Power of Money, its Determination and Relation to Credit, Interest and Crises*. Indianapolis, IN, USA : The Online Library of Liberty, 1911

[Guntsch et al. 2001] Guntsch, Michael ; Middendorf, Martin ; Schmeck, Hartmut: An Ant Colony Optimization Approach to Dynamic TSP. (2001), S. 860867. ISBN 1558607749

[Horoba and Sudholt 2010] Horoba, Christian ; Sudholt, Dirk: Ant Colony Optimization for Stochastic Shortest Path Problems. (2010), S. 14651472. ISBN 9781450300728

[Hsu 2014] Hsu, Chih-Ming: An integrated portfolio optimisation procedure based on data envelopment analysis, artificial bee colony algorithm and genetic programming. In: *International Journal of Systems Science* 45 (2014), Nr. 12, S. 2645–2664

[Huang and Liao 2008] Huang, Kuo-Ling ; Liao, Ching-Jong: Ant colony optimization combined with taboo search for the job shop scheduling problem. In: *Computers Operations Research* 35 (2008), Nr. 4, S. 1030–1046. – ISSN 0305-0548

[Huggett 1993] Huggett, Mark: The risk-free rate in heterogeneous-agent incomplete-insurance economies. In: *Journal of Economic Dynamics and Control* 17 (1993), Nr. 5, S. 953–969. – ISSN 0165-1889

[Janner et al. 2019] Janner, Michael ; Fu, Justin ; Zhang, Marvin ; Levine, Sergey: When to trust your model: Model-based policy optimization. In: *arXiv preprint arXiv:1906.08253* (2019)

[Kumar et al. 2009] Kumar, Sameer ; Chadha, Gitika ; Thulasiram, Ruppa K. ; Thulasiraman, Parimala: Ant Colony Optimization to price exotic options. (2009), S. 2366–2373

[Lin et al. 2013] Lin, William T. ; Tsai, Shih-Chuan ; Lung, Pei-Yau: Investors' herd behavior: Rational or irrational? In: *Asia-Pacific Journal of Financial Studies* 42 (2013), Nr. 5, S. 755776

[Lucas 1981] Lucas, Robert E.: *Rational Expectations and Econometric Practice: Volume 1.* NED - New edition. University of Minnesota Press, 1981

[McCall 1970] McCall, J. J.: Economics of Information and Job Search. In: *The Quarterly Journal of Economics* 84 (1970), Nr. 1, S. 113–126

[Muth 1961] Muth, John F.: Rational Expectations and the Theory of Price Movements. In: *Econometrica* 29 (1961), Nr. 3, S. 315–335. – ISSN 00129682, 14680262

[Sutton 1991] Sutton, Richard S.: Dyna, an integrated architecture for learning, planning, and reacting. In: *ACM Sigart Bulletin* 2 (1991), Nr. 4, S. 160–163

[Tversky and Kahneman 1982] Tversky, Amos ; Kahneman, Daniel: Judgment under uncertainty: Heuristics and Biases. In: *Judgment under Uncertainty* (1982), S. 320

[Zheng et al. 2020] Zheng, Stephan ; Trott, Alexander ; Srinivasa, Sunil ; Naik, Nikhil ; Gruesbeck, Melvin ; Parkes, David C. ; Socher, Richard: The AI Economist: Improving Equality and Productivity with AI-Driven Tax Policies. (2020)

APPENDIX

A1. Environment Parameters

	McCall		Huggett
Max Wage	60	Max Assets	20
Unemployment Benefits	25	Max Debt	5
—	—	Interest Rate	0.01
—	—	CRRA Gamma	1.5
Discount Rate (γ)	0.9	—	0.5

A2. Optimizer Parameters

	Q-Learning	DynaQ	DynaQ+	ACO DynaQ+
Learning Rate (α)	0.1	0.1	0.1	0.1
Exploration Rate (ϵ)	0.3	0.1	0.1	0.1
Planning Steps	—	50	50	50
Ants	—	—	—	50

A3. Model Parameters

	Vanilla Dyna	Time Dyna	ACO Dyna
Time Weight (κ)	—	1×10^{-4}	1×10^{-4}
Pheromone Weight (α)	—	—	0.5
Belief Weight (β)	—	—	0.5
Evaporation Rate (ν)	—	—	0.1