The Coconut Model with Adaptive Strategies

S. Banisch, E. Olbrich
Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany

Abstract

We continue our development of an agent-based implementation of the Coconut Model by Peter Diamond and report on the achievements. While dealing with fixed but heterogeneous strategies in previous work we now concentrate on learning dynamics. Namely, we introduce temporal difference learning as a way to procedurally solve the optimization problem as posed in the original paper. We show that the model with this kind of adaptive agents converges to a considerable degree to the original theoretical results for an infinite and homogeneously adapting population. Conclusions regarding non-equilibrium trajectories and equilibrium selection can be drawn from that. Having an agent-based baseline well established, we introduce a model extension with two goods that must be combined to a third elaborate product for consumption. First experiments reveal that rich behavioral regimes emerge in such a setting.

1 Introduction

Imagine an island with N agents that like to eat coconuts. They search for palm trees and harvest a nut from it if the tree is not too tall, meaning that its height does not exceed an individual threshold cost ($c_{tree} < c_i$). However, in order to consume the nut and derive utility y from this consumption agents have to find a trading partner, that is, another agent with a nut. Therefore, the agents have to base their harvest decision *now* (by setting c_i) on their expectation to find a trading partner *in the future*. Or, less metaphorically, agents are faced with production decisions that have to be evaluated based on their expectations about the future utility of the produced entity which in turn depends on the global production level via a trading mechanism. For this reason, the Coconut Model is useful not only for the incorporation of heterogeneity [2], but also for the analysis of adaptive agents that – rationally or not – have to form expectations about the future system state in order to evaluate their decision options.
In the original papers [6, 5] this problem of inter-temporal optimization is formulated using dynamic programming principles and the Bellmann equation in particular. The author(s) arrive at a differential equation (DE) that describes the evolution of the cost threshold along an optimality path (where the individual thresholds are all equal $c_i = c$) which is coupled to a second DE describing the evolution of the number of coconuts in the population. However, knowing the optimal dynamics, that is, the differential equations that an optimal solution has to fulfill, is not sufficient to study problems such as equilibrium selection or stability in general, because the optimality conditions do not say anything about the behavior of the system when it is perturbed into a suboptimal state. On the other hand, the Bellmann equation is also at the root of reinforcement learning algorithms and temporal difference (TD) learning in particular which are known to converge to this optimality under certain conditions [13]. The incorporation of learning

The work was supported from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318723 (MatheMACS). S.B. also acknowledges financial support by the Klaus Tschira Foundation. The first part of this work will appear in JASSS, special section for *Artificial Economics* 2015 [3].

in the agent-based version of the Coconut Model and the assessment of its adequacy by comparison to the original solution is the main contribution of this paper.
The necessity to take into account not only the result of rational choice but to focus more on the processes that may lead to it has been pointed at by Simon almost 40 years ago [12]. Around ten years later, the notion of artificial adaptive agents has been proposed by Holland and Miller [8] who define an *adaptive* agent by two criteria: (1.) the agent assigns a value (fitness, accumulated reward, etc.) to its actions, and (2.) the agent intends to increase this value over time (p. 365). Virtually all models with adaptive agents proposed since then follow these principles. In genetic algorithms, for instance, an evolutionary mechanism is implemented by which least fit strategies are replaced by fitter ones and genetic operators like recombination and mutation are used to ensure that potential improvements are found even in high-dimensional strategy spaces (e.g., [7, 14]). Another approach which became prominent during the last years could be referred to as strategy switching (e.g., [1, 4, 9, 10]). Here agents constantly evaluate a set of predefined decision heuristics by reinforcement mechanisms and chose the rule that performs best under the current conditions.
The TD approach used here differs mildly from these models but fits well with the abstract specification of adaptive behavior proposed in [8]. In our case agents learn the value associated to having and not having a coconut in form of the expected future reward and use these values to determine their cost threshold c_i. That is, agents are forward-looking by trying to anticipate their potential future gains. While checking genetic algorithms or strategy switching methods in the context of the Coconut Model is an interesting issue for future research, in this paper we would first like to derive an agent-based version of the model that is as closely related to the original model as possible.

2 Individual Choice the Original Model

In previous work [2], we have concentrated on the state dynamics and the effect of heterogeneous but fixed individual strategies. For the homogeneous case, the state dynamic is given by

$$\dot{\epsilon} = f(1-\epsilon)G(c) - \epsilon^2 \tag{1}$$

where ϵ is the ratio of agents having a coconut, f the probability that agents find a coco tree and $G(c)$ a cumulative distribution defining the probability that the cost of a tree is smaller than the cost accepted by the agents $c_{tree} < c$. We gave a detailed account of how to incorporate heterogeneous cost thresholds c_i into (1). (See [2] for the details.)
A crucial ingredient of the coconut model, however, is that agents are not allowed to directly consume the coconut they harvested. They rather have to search for a trading partner, that is, for another agent that also has a coconut. The idea behind this is that agents have to find buyers for the goods they produce. If an agent that possesses a nut encounters another agent with a nut both of them are supposed to consume instantaneously and derive each a reward of y from this consumption. In effect, this means that the expected value of climbing a tree depends on the total number of coconuts in the population or, more precisely, on the time agents have to wait until a trading partner will be found. Rational agents are assumed to maximize their expected future utility

$$V_i(t) = \mathbb{E} \int_t^\infty e^{-\gamma(\tau - t)} r_i(\tau) d\tau$$

where $r_i(\tau)$ corresponds to the cost of climbing or respectively to the utility y from consumption of agent i at time τ and γ to the discount factor. A fully rational agent has to find the strategy c_i^* that maximizes its expected future reward and, since agents cannot consume their coconut instantaneously, this reward depends on his expectation about their trading chances. This can be formulated as a dynamic programming problem with $dV_i(t)/dt = -\mathbb{E}[r_i(t)] + \gamma V_i(t)$. Considering that there are two states (namely, $s_i = 0$ or $s_i = 1$) there is a value associated to having ($V_i(s_i = 1, t) := V_i^t(1)$) and to not having ($V_i(s_i = 0, t) := V_i^t(0)$) a coconut at time t. As a rational agent accepts any opportunity that increases expected utility, a necessary condition for an optimal strategy is $c_i^* = V_i^t(1) - V_i^t(0)$. By this reasoning, assuming homogeneous strategies $c_i^* = c^*$ Diamond derives another DE that describes the evolution of the optimal strategy

$$\frac{dc^*}{dt} = \gamma c^* + \epsilon(c^* - y) + f\left[\int_0^{c^*} (c^* - c) dG(c)\right] \tag{2}$$

The fixed point solutions of the model are then given by the points (ϵ^*, c^*) for which (1) and (2) are zero.

3 Individual Choice by Temporal Difference Learning

We shall now turn to an adaptive mechanism by which the strategies are endogenously (and heterogeneously) set by the agents. As in [2], we follow in this implementation the conception of Diamond [6] as closely as possible. That is, firstly, the threshold c_i has to trade off the cost of climbing against the expected future gain of earning a coconut from it. In other words, agents have to compare the value (or expected performance if you wish) of having a coconut $V_i^t(1)$ with the value $V_i^t(0)$ of staying without a nut. If the difference between the expected gain from harvesting at time t and that of not harvesting $(V_i^t(1) - V_i^t(0))$ is larger than the cost of the tree c_{tree}, agents can expect a positive reward from harvesting a nut now. Therefore, in accordance to [6], it is reasonable to set $c_i^t = V_i^t(1) - V_i^t(0)$.
Now, how do agents arrive at reliable estimates of $V_i^t(1)$ and $V_i^t(0)$? We propose that they do so by a simple temporal difference (TD) learning scheme that has been designed to solve dynamic programming problems as posed in the original model. Notice that for single-agent Markov decision processes temporal difference schemes are proven to converge to the optimal value functions [13]. In the Coconut Model with agents updated sequentially it is reasonable to hypothesize that we arrive at accurate estimates of $V_i^t(1)$ and $V_i^t(0)$ as well. Notice that agents do not learn the ϵ–dependence explicitly, but condition their actions only on their own current state. As a result, agents will only learn optimal stationary strategies. The consideration of more complex (and possibly heterogeneous) information sets, including previous trends and information about other agents might lead to a richer set of solutions and points to interesting extensions of the model. However, we think that it is useful to first understand the basic model and relate it to the available theoretical results as this will also be needed to understand additional contributions by model extensions.

3.1 Learning the Value Functions by Temporal Differences

The learning algorithm we propose is a very simple value TD scheme. Agents use their own reward signal r_i^t to update the values of $V_i^t(s_i^t = 1)$ and $V_i^t(s_i^t = 0)$ independently

from what other agents are doing. In each iteration agents compute the TD error by comparing their current reward plus the discounted expected future gains to their current value estimate

$$\delta_i^{t+1} = \underbrace{r_i^{t+1}}_{\text{reward}} + \underbrace{e^{-\gamma N^{-1}}\left[s_i^{t+1}V_i^t(1) + (1-s_i^{t+1})V_i^t(0)\right]}_{\text{estimated discounted future value}} - \underbrace{\left[s_i^t V_i^t(1) + (1-s_i^t)V_i^t(0)\right]}_{\text{current estimate}}. \quad (3)$$

Notice that the discount factor γ as defined for the time continuous DE system is rescaled as $\gamma_r = e^{-\gamma N^{-1}}$ for the discrete-time setting and in order to account for the finite simulation with asynchronous update in which only one (out of N) agents is updated in each time step (N^{-1}). The iterative update of the value functions is then given by

$$\begin{aligned} V_i^{t+1}(1) &= V_i^t(1) + \alpha\delta_i^{t+1}s_i^t \\ V_i^{t+1}(0) &= V_i^t(0) + \alpha\delta_i^{t+1}(1-s_i^t) \end{aligned} \quad (4)$$

such that $V_i(1)$ ($V_i(0)$) is updated only if agent i has been in state 1 (0) in the preceding time step.

The idea behind this scheme and TD learning more generally is that the error between subsequent estimates of the values is reduced as learning proceeds which implies convergence to the true values. The form in which we implement it here is probably the most simple one which does not involve update propagation using eligibility traces usually integrated to speed up the learning process [13]. In other words, agents update only the value associated with their current state s_i^t. While simplifying the mathematical description (the evolution depends only on the current state) we think this is also plausible as an agent decision heuristic.

All in all the model implementation[1] is

(0) Initialization: set initial values $V_i^0(1), V_i^0(0)$ and states s_i^0 according to the desired initial distribution. Set initial strategies $c_i^0 = V_i^0(1) - V_i^0(0)$.
(1) Iteration loop I (search and trade):
  (a) random choice of an agent i with probability $\omega(i) = 1/N$
  (b) `if` $s_i = 0$ climb a coco tree with probability $fG(c_i)$ and harvest a nut, i.e., $s_i^{t+1} = 1$
  (c) `else` trade (consume) with probability ϵ such that $s_i^{t+1} = 0$
(2) Iteration loop II (learning):
  (a) compute TD error δ_i^{t+1} for all agents with reward signal $r_j = 0, \forall j \neq i$ and r_i depending on the action of i in part (1)
  (b) update relevant value function by $V_i^{t+1}(s_i^t) = V_i^t(s_i^t) + \alpha\delta_i^{t+1}$ for all agents
  (c) update strategy by $c_i^{t+1} = V_i^{t+1}(1) - V_i^{t+1}(0)$

If not stated otherwise, the simulation experiments that follow are performed with the following parameters. The costs of trees are uniformly drawn from the interval defined by $c_{max} = 0.5$ and $c_{min} = 0.3$. A strategy c_i larger than c_{max} hence means that the agent accepts any tree, $c_i < c_{min}$ that no tree is accepted at all. The rate of tree encounter is $f = 0.8$ and the utility of coconuts is $y = 0.6$. We continue considering a relatively small system of 100 agents and the learning rate is $\alpha = 0.05$. The parameter γ is the discount rate with small values indicating farsighted agents whereas larger values discount

[1] See `www.openabm.org/model/5045` for a MatLab implementation made available on the OpenABM archive.

future observations more strongly. The system is initialized (if not stated otherwise) with $\epsilon^0 = 0.5$, $V_i^0(1) = y$ and $V_i^0(0) = 0$ for all agents such that $c_i^0 = y > c_{max}$ and everybody climbs in the beginning.

3.2 Convergence Behavior with Learning

We now report on the overall convergence behavior of the ABM as a function of γ and compare it to the fixed point solution of [6], see also [11]. There are two interesting questions here: (1.) what happens as we reach the bifurcation value $\gamma > \gamma^*$ at which the two fixed point curves $\dot{\epsilon} = 0$ and $\dot{c} = 0$ cease to intersect? (2.) in the parameter space where they intersect, which of the two solutions is actually realized by the ABM with TD learning?

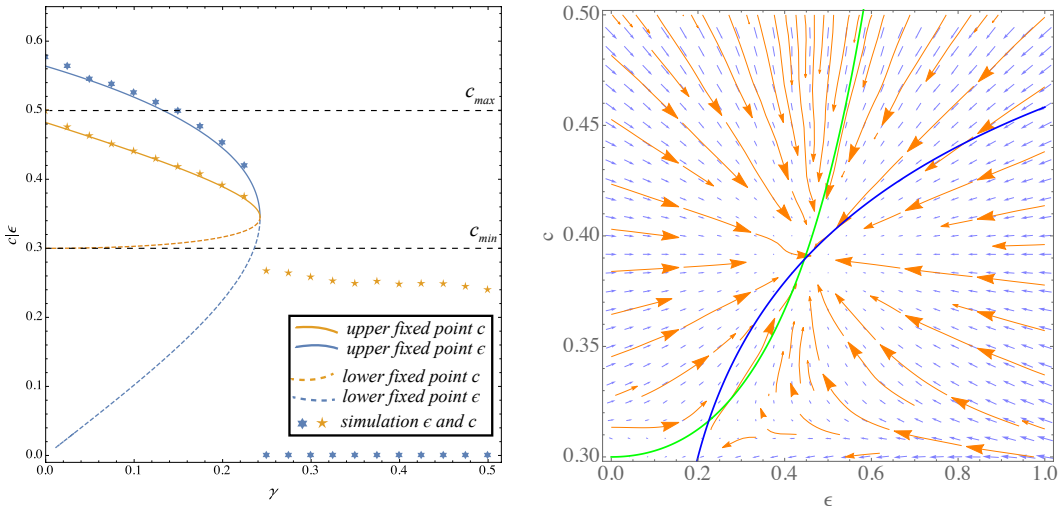

Fig. 1. L.h.s.: The fixed point behavior of the DE system (1) - (2) for $\gamma \in [0, 0.5]$ is compared to single model realizations (200000 steps) for different γ. The agent model with TD learning scheme converges closely to the upper theoretic fixed point values. R.h.s.: Numerically computed vector field for the dynamics of the agent model for $\gamma = 0.2$. The fixed point curves of the DE system are also shown.

Both questions are answered with Fig. 1. First, if γ becomes large, the ABM converges to the state in which agents do not climb any longer. That is, $\epsilon^* = 0$ and $c^* < c_{min}$. However, additional simulation experiments showed that the bifurcation takes place at slightly lower values of γ. In fact, these experiments revealed that the learning rate α governing the fluctuations of the value estimates plays a decisive role (the larger α, the smaller the bifurcation point). See [3] for some more details. Besides these small deviation, however, Fig. 1 shows that on the whole the ABM reproduces the theoretical results with considerable accuracy.

Regarding the second question – that is, equilibrium selection – Fig. 1 provides strong indications that the only stable solution for the simulated dynamics is the upper fixed point, sometimes referred to as »optimistic« solution. The vector field (numerically computed) on the r.h.s. renders visible that the lower fixed point acts as a saddle under the learning dynamics. Depending on the initial strategies and coconut level, when close to the »pessimistic fixed point« the system is driven either to the »optimistic« solution or to a no-production state.

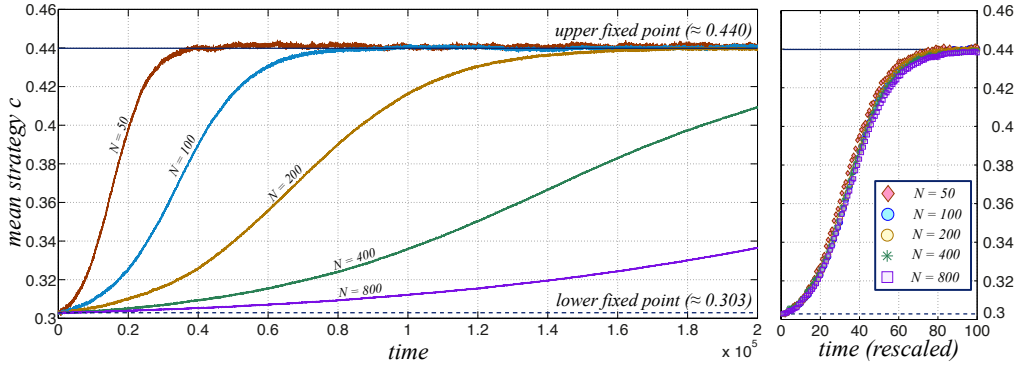

Fig. 2. R.h.s.: Time evolution of the system initialized at the low fixed point (dashed dark line) for different system sizes. L.h.s.: The same learning curve is obtained when rescaling time by the number of agents.

We will confirm this by providing numerical arguments for the instability of the »pessimistic solution« initializing the model at that point. We concentrate again on the parameterization used in the previous sections with $f = 0.8, y = 0.6, c_{min} = 0.3, c_{max} = 0.5$, climbing costs uniformly distributed in $[c_{min}, c_{max}]$ and stick to a discount rate $\gamma = 0.1$. Fig. 2 shows the evolution (200000 steps) of the ABM with TD learning for an initialization at the low fixed point (shown by the dashed dark line). There are 100 agents and the learning rate is $\alpha = 0.025$. Each curve in the plot is an average over 5 simulation runs.
It becomes clear that trajectories are repelled from the low fixed point into the direction of the »optimistic« solution. As the size of the system increases, the initial period in which the system stays close to the lower fixed point increases. However, as shown on the right of Fig. 2, the differences between the learning curves in systems of different size vanish when time is rescaled by the number of agents such that one time step accounts for N individual updates. This provides further evidence for the instability of the lower fixed point which we cannot expect to become stable in the large (infinite) system.

4 A Model with Three Products

The analyses we have performed so far, document the development of an agent-based baseline model that matches with the behavior of Diamond's theoretical model from the 1980ies. Having that baseline well-understood, there are now many ways in which the agent version may be extended so to create an artificial economy with more complex ingredients. Here we present results for a model in which agents produce two basic goods which must be combined to a third one in order to derive utility from consumption. These two basic goods however may be sold to or bought from others which increases the number of decision alternatives.[2]
We keep the setup very close to the model analyzed in the previous sections. There are now two basic products (or resources) A and B which agents produce if the respective cost threshold c_A, c_B is above a randomly drawn production cost (for A and B this cost drawn uniformly from $[c_{min}, c_{max}]$). However, agents cannot consume these basic products anymore (as opposed to the simple coconut setting). They have to gather both basic goods (say coconuts and bananas) to »produce« an elaborate product AB (say a coco-banana shake). We put »produce« in parenthesis because here we simply assume

[2] This model extension will also be made available on OpenABM before the conference.

that this production is accomplished as soon as the two products have been gathered. More complex variants including production costs and times associated to that step are highly encouraging but omitted in this first analysis which is aimed at understanding the basic phenomena that can occur in such a setting. In the model, only AB in conjunction are consumed on encountering another agents that possess AB as well. That is, only if two AB-agents meet they obtain reward y by consuming their coco-banana shake.
However, agents are now also allowed to buy and sell the two basic products A and B to others. This becomes an additional source of deriving utility from production which has not been present in the original model. For simplicity we assume that climbing costs, prices paid when selling a basic good as well as utility from consumption have the same unit and are accounted as positive (sell, consume) or negative (produce, buy) rewards. Here we assume a price formation process based on the values the trading partners have learned for the different states ($s_i \in \{0, A, B, AB\}$). Other mechanisms such as markets are a compelling future ingredient as well.

4.1 Model

Values. We extend the value learning scheme analyzed above in the one-product case in a straightforward way to three products. That is, each agent holds and updates four values one for each possible state $V_i(s) : s \in \{0, A, B, AB\}$. We explain their updating below.
Prices. We treat prices as differences in values that agents compute on the basis of their value functions. In analogy to the original model, the cost threshold (now referred to as price) $p_i^A(0 \to A) = V_i(A) - V_i(0)$ specifies the price agent i would be willing to pay in order to harvest an A-tree. Additionally $p_i^A(0 \to A)$ now also defines the price i would maximally pay when offered A by another agent. The price $p_i^B(0 \to B)$ is given and used equivalently. There are two more value differences that play a role in our model, namely $p_i^A(B \to AB)$ and $p_i^B(A \to AB)$. They define what agent i is willing to pay for $B(A)$ if (s)he already possesses $A(B)$ and would have AB after buying. They are given by

$$p_i^A(B \to AB) = V_i(AB) - V_i(B) \tag{5}$$
$$p_i^B(A \to AB) = V_i(AB) - V_i(A) \tag{6}$$

Thus, price formation is based on the expected gain in value associated to the state-transition the respective action leads to.
Basic Production. Production is conceived as before and for the two raw materials symmetrically. Agents encounter coco (A) and banana (B) trees at a fixed and homogeneous rate $f_A = f_B = f$. The costs of the trees are equally distributed as well, that is: uniformly in $[c_{min}, c_{max}]$. Agents climb an occurring tree if the *price* (given by the respective value difference, see above) they would be willing to pay exceeds that randomly drawn cost ($c_{tree} < c_i$). They have to pay that cost (c_{tree}) for production. For instance, consider that an agent that has B encounters an A–tree. Then the cost threshold (the price (s)he is willing to pay) is given by $p_i^A(B \to AB) = V_i(AB) - V_i(B)$. The agent will climb if $p_i^A(B \to AB)$ is above the randomly drawn cost c_{tree}. This cost will be i's (negative) reward r_i^{t+1} for that time step.
Consumption. We assume, that in the three-product scenario only the elaborated good AB can be consumed by trading with others. Therefore, if agent i meets another agent j such that both hold AB both instantaneously consume the elaborated good and derive

utility y from it. A and B alone cannot be consumed in trade but be sold or bought as specified below.

Trading. Agents are allowed to sell the raw materials A and B they hold to others. Say agent i possesses A and is chosen for interaction with another agent j. As a first case assume that $s_j = 0$, that is, agent j has no good at the moment. The price that j is willing to pay for A is $p_j^A(0 \to A) = V_j(A) - V_j(0)$. Conversely, the price i assigns to A is given by $p_i^A(0 \to A) = V_i(A) - V_i(0)$. Agent i will sell A if $p_j^A(0 \to A) > p_i^A(0 \to A)$ at the intermediate price

$$p^A = \frac{p_j^A(0 \to A) + p_i^A(0 \to A)}{2}. \tag{7}$$

This procedure mimics an idealized bargaining process in which the values p_j^A and p_i^A are first used to assess whether there is an interest in trading at all and then form the basis for repeated bids during price negotiation. If i sells A to j, agent i receives reward $r_i^{t+1} = p^A$ and j reward $r_j^{t+1} = -p^A$ during that round while $s_i^{t+1} = 0$ and $s_j^{t+1} = A$. As a second case consider j already possesses B. The agent would then be willing to pay $p_j^A(B \to AB) = V_j(AB) - V_j(B)$ for A which may become even higher if j has learned that consumption of AB pays. In the other cases, $s_j = A$ or $s_j = AB$, A is not sold.

Value update. For the update of the value functions for each agent the temporal difference error is computed as

$$\delta_i^{t+1} = r_i^{t+1} + e^{-\gamma N^{-1}} V_i^t(s_i^{t+1}) - V_i^t(s_i^t) \tag{8}$$

where $r_i^{t+1} \neq 0$ only if i has performed any action (associated to a reward) during that step, $e^{-\gamma N^{-1}} V_i^t(s_i^{t+1})$ the discounted future value estimated on the basis of the new state and $V_i^t(s_i^t)$ the value associated to the old state (notice that $s_i^t = s_i^{t+1}$ for all agents that made no action at t). The values are

$$V_i^{t+1}(s) = \begin{cases} V_i^t(s) + \alpha \delta_i^{t+1} & \text{if } s \equiv s_i^t \\ V_i^t(s) & \text{else} \end{cases} \tag{9}$$

such that only the value of the good with which agent i entered the iteration step is evaluated.

Event Schedule. In the model, we first chose two agents (i and j) at random. At the maximum, one (trans)action is performed during each time step and the different options are tried in the following order: 1. consumption if $s_i = s_j = AB$; 2. i tries to sell to j if $s_i \neq 0$; 3. i performs the production step if a tree is occurring and the cost is lower than the respective price. Notice that the order of attempts for different actions may have a strong impact on the model behavior and should in a more realistic setting be something agents decide about.

Parameters and Initialization. If not otherwise stated, the frequency with which coco and banana trees occur are $f_A = f_B = 0.4$ and the cost distributed uniformly in $[c_{min}, c_{max}] = [0.3, 0.5]$. The reward on consumption is $y = 4.0$. The learning rate is $\alpha = 0.1$ and the continuous time discount factor $\gamma = 0.1$ or $\exp(-\gamma/N) = 0.999$ for the discrete system of 100 agents. Agents are chosen with equal probability and there are no geographical constraints.

In all simulations performed here, there are no products in the population at $t = 0$, i.e. $s_i = 0 : \forall i \in N$. The initial values $V_i^0(A), V_i^0(B), V_i^0(AB)$ are assigned at random uniformly in $[c_{min}, c_{min} + y(c_{max} - c_{min})]$ except for $V_i^0(0)$ which is set to zero.

4.2 Time Evolution

In Fig. 3, the time evolution of the model is shown for a system of 100 agents. The 100000 first time steps are shown in which on average each agent has been chosen 1000 times. Out of a situation without any goods in the population, we observe the fast accumulation of coconuts (A) and bananas (B) in the population. This is followed by a slow increase of the number of AB–agents. The number of agent without any good also increases again as these agents are more likely to encounter partners for consumption. Eventually, this realization settles at a state where approximately 30% of the agents have no goods at their disposal, 50 % hold either A or B and 20 % possess AB.

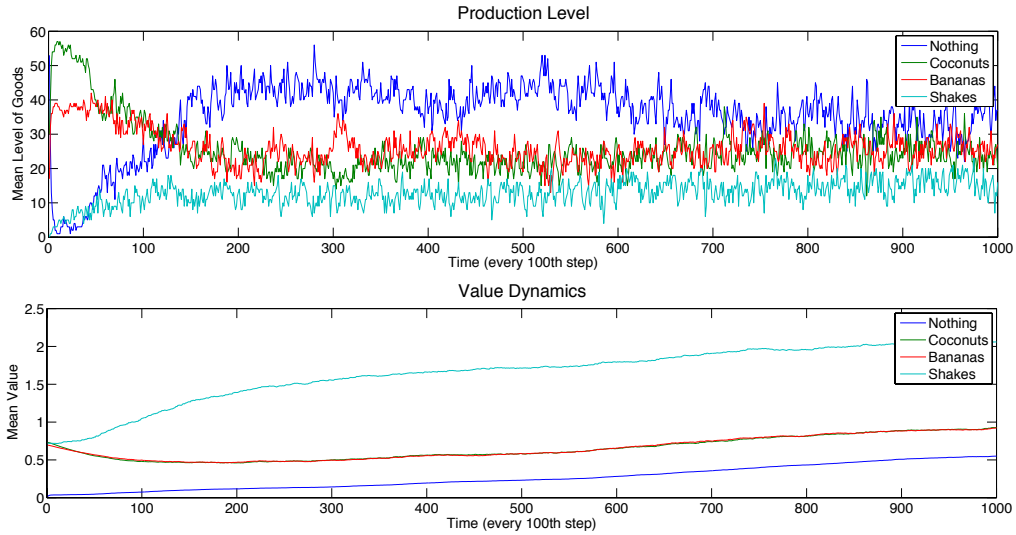

Fig. 3. The first 100000 steps of a simulation of 100 agents are shown.

The plot on the bottom of Fig. 3 shows the evolution of the mean values agents assign to the four different states. Not surprisingly, the value estimates of A and B behave in exactly the same way, settling at around 1.6 in the long run of the model ($\approx$ 300000 steps). The value for AB is highest and settles at around 2.8 in the this parameter constellation, and the value of having no good settles at around 1.2. Notice that the prices (and threshold for production respectively) are given by the differences in values which converge much faster. With this constellation of values, on average, agents production threshold ($p_i^A(0 \to A)$ and $p_i^B(0 \to B)$) is around 0.4 just in between c_{min} and c_{max}. On the other hand, when already possessing of one basic good, agents are willing to pay $p_i^A(B \to AB) \approx 1.2$ for the good that is missing for a coco-banana shake (AB).

4.3 Behavioral Regimes

These average values and prices however do not reveal that an interesting differentiation of agent behavior emerges in the simulation. Namely, there are three different behavioral strategies which different agents learn in the interaction with this artificial economy. This is shown in Fig. 4.
Let us first consider the right-hand side of that figure which shows all agents in a plane spanned by the number of buying versus selling actions performed in 100000 time steps

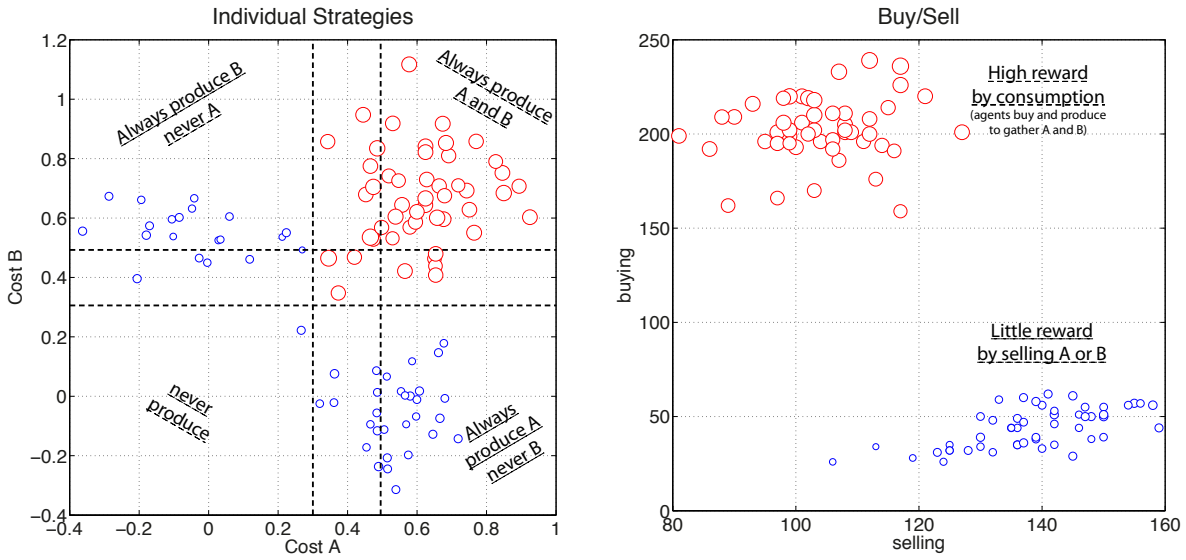

Fig. 4. Distribution of agents in the plane spanned by production costs $p_i^A(0 \to A)$ and $p_i^B(0 \to B)$ (l.h.s) and selling versus buying actions (r.h.s.) in a simulation of 100 agents during 100000 steps (≈ 1000 actions for each agent). The circle sizes represent the accumulated rewards. Two different behavioral regimes emerge as values approach an heterogeneous pattern in which some agents learn to produce only one basic product.

(again 1000 steps on average for each single agent). The sizes of the data points represents the accumulated reward of the respective agent. We clearly observe two groups of agents: one which performs very well and is buying approximately two times more compared to selling actions (red circles); and a second one which makes less (but positive) reward by selling the raw products they produced (blue circles).
The left-hand side of Fig. 4 shows how this behavioral distinction is reflected in the values different agents have learned for the different goods. Here, the production thresholds $p_i^A(0 \to A)$ and $p_i^B(0 \to B)$ are plot against one another. While the red well-performing agents tend to produce the two products by assigning a relatively high value to A and B, the other group represented by the blue circles associates a low (even negative) value to one of the raw materials. That is, the respective value is below the minimum cost of trees (dashed lines) and they never produce that good. Consequently, the only way in which these agent gather reward is by selling the raw material they »specialized on« to agents from the other group that, conversely, treat the former as an additional source of raw material supply.

4.4 Restricted Production

As a further example we look at the case that half of the population is not capable of any production. This is modeled by letting $f_A = f_B = 0$ for one half of the agents referred to as non-producers. The question is if these agents are capable of using the option to buy products from the producers in order to accumulate reward by consumption and how they achieve that. This is shown in Fig. 5 for 200 agents where non-producers are represented by blue circles and producers plot in red.
Again we observed the emergence of different behavioral strategies triggered by the fact that different agents learn different values for the four possible states. First, the group of producers (red circles) differentiates as before into »sellers« and »consumers« with the

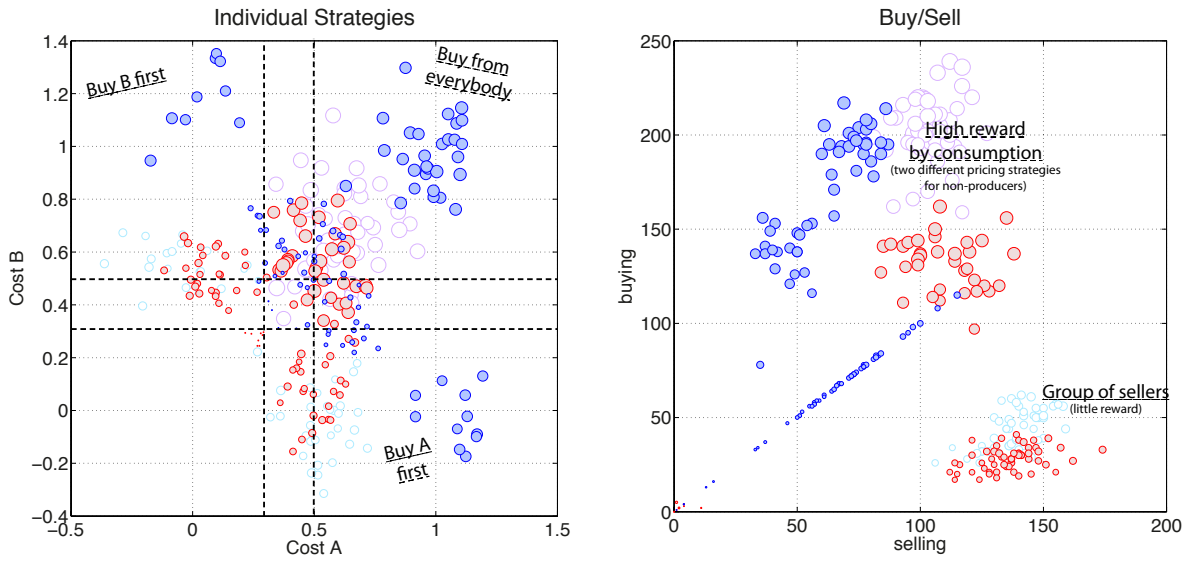

Fig. 5. Distribution of agents in the plane spanned by production costs $p_i^A(0 \to A)$ and $p_i^B(0 \to B)$ (l.h.s) and selling versus buying actions (r.h.s.). 200 agents are considered divided into producers (red) and non-producers (blue). Sizes of the circles represent the accumulated reward during 200000 steps ($\approx$ 1000 actions for each agent).

former accumulating high rewards compared to the latter. The group of non-producers, moreover, differentiates into four different sub-groups. First, some of these agents assign a very high price to both $p_i^A(0 \to A)$ and $p_i^B(0 \to B)$ meaning that they are willing to buy from everybody. These agents manage to perform very well and almost as good as the well–performing producers. A second group of agents shows very little economic activity and sells all the products bought before (and is therefore located along the diagonal on the right-hand plot). The other two sub–groups among the non–producers develop a rather sophisticated trading strategy by converging to a value difference that is very high for one but low for the other basic good. When without any good, they will buy one basic good first (say B and compare with the augmented description in Fig. 5) and never buy the other one. Only when the first basic good is at their disposal, they will purchase the other one to obtain and finally consume AB.

4.5 Discussion

The reason for these different behaviors to emerge in the simulation is related to the fact that not all agents do explore the entire set of possible actions. Most importantly, the group of »sellers« never experiences the high reward that is possible by consumption of AB and therefore never learns that the value for having both basic goods at the same time is actually high. Noteworthy, this effect is stable even with a considerable amount of exploration implemented by adding a noise term to the value estimates.

From the point of view of multi-agent learning this effect may be read as a deficit of the learning scheme incapable of ensuring convergence to the optimal solution. From the point of view of agent-based modeling, on the other hand, it appears not completely implausible that some agents experience a sequence of events related to low but positive rewards that forces them into a behavioral regime in which some options are no longer accessible. In this case, suboptimal behavior may lead to interesting constellations of agent behaviors that sustain one another, as exemplified in the previous section.

5 Summary

In the first part of this contribution we continued the development of a theory-aligned agent-based version of Diamond's coconut model [6]. In the model agents have to make investment decisions to produce some good and have to find buyers for that good. Step by step, we analyzed the effects of single ingredients in that model – from homogeneous to heterogeneous (presented last year [2]) to adaptive strategies (this paper) – and relate them to the qualitative results obtained from the original dynamical systems description. We computationally verify that the overall behavior of the ABM with adaptive strategies aligns to a considerable accuracy with the results obtained in the original model. The main outcome of this exercise is the availability of an abstract baseline model for search equilibrium which allows to analyze more realistic behavioral assumptions such as trade networks, heterogeneous information sets and different forms of bounded rationality but contains the idealized solution as a limiting case.

Another contribution is the incorporation of temporal difference (TD) learning as a way to address problems that involve inter-temporal optimization in an agent-based setting. The coconut model serves this purpose so well because the strategy equation in the original paper is based on dynamic programming principles which are also at the root in this branch of reinforcement learning. Due to this common foundation we arrive at an adaptive mechanism for endogenous strategy evolution that converges to one of the theoretical equilibria, but provides, in addition to that, means to understand how (and if) this equilibrium is reached from an out-of-equilibrium situation. Such a characterization of the model dynamics is not possible in the original formulation.

Finally, we presented a model extension to three products. To stay with the island metaphor of Diamond, agents now search for coco and banana trees and produce a coco-banana shake once the two basic goods are at their disposal. Only these shakes are consumed on encountering another agent. But agents are now also allowed to sell their fruits. This model gives rise to interesting constellations of heterogeneous behaviors compelling enough for future examination.

1. W. B. Arthur, *Inductive reasoning and bounded rationality*, The American economic review, 84 (1994), pp. 406–411.
2. S. Banisch and E. Olbrich, *The Diamond Model with Heterogeneous Agent Strategies.* 11th Artificial Economics Conference, Porto, Portugal, 2015.
3. S. Banisch and E. Olbrich, *The Coconut Model with Heterogeneous Strategies and Learning*, Journal of Artificial Societies and Social Simulation, (in press).
4. W. A. Brock and C. H. Hommes, *A rational route to randomness*, Econometrica: Journal of the Econometric Society, (1997), pp. 1059–1095.
5. P. Diamond and D. Fudenberg, *Rational expectations business cycles in search equilibrium*, Journal of political Economy, (1989), pp. 606–619.
6. P. A. Diamond, *Aggregate demand management in search equilibrium*, The Journal of Political Economy, (1982), pp. 881–894.
7. J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.*, U Michigan Press, 1975.
8. J. H. Holland and J. H. Miller, *Artificial adaptive agents in economic theory*, The American Economic Review, 81 (1991), pp. 365–370.
9. C. Hommes, *Behavioral rationality and heterogeneous expectations in complex economic systems*, Cambridge University Press, 2013.
10. S. Landini, M. Gallegati, and J. E. Stiglitz, *Economies with heterogeneous interacting learning agents*, Journal of Economic Interaction and Coordination, 10 (2015), pp. 91–118.
11. T. Lux, *A note on the stability of endogenous cycles in diamond's model of search and barter*, Journal of economics, 56 (1992), pp. 185–196.
12. H. A. Simon, *Rationality as process and as product of thought*, The American economic review, 68 (1978), pp. 1–16.
13. R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, MIT press, 1998.
14. N. J. Vriend, *An illustration of the essential difference between individual and social learning, and its consequences for computational analyses*, Journal of economic dynamics and control, 24 (2000), pp. 1–19.