# Q-Learning in Unstable Relationships

Ming Kang

November 2005
VERY PRELIMINARY – COMMENTS WELCOME

**Abstract**

This study revisits Burdett, Imai and Wright's model of endogeneously unstable relationships [1]. In BIW, bilateral relationships are characterized as stable or unstable according to whether matched agents continue to search for new partners. Endogenous instability hinges on mutually reinforced beliefs about the search and acceptance behaviour of match partners in equilibrium. The model predicts multiple rational expectations (RE) equilibria in some instances and a unique RE equilibrium in others. Here, I investigate to what extent these strong equilibrium predictions are borne out when individuals are modelled as adaptive learning agents with no *a priori* knowledge of the environment. Attention is restricted to the version of the BIW model with just two match types and a break-up rule that precludes returning to one's previous partner once a new partner is found.

Watkin's [9] Q-learning algorithm is used to simulate the adaptive learning process in the BIW environment. Consistent with the psychological concept of reinforcement, agents make subjective evaluations of behavioural alternatives based solely on direct experience and higher-valued alternatives are selected more often than lower-valued ones. Simulations of the Q-learning process point to cases in which the equilibrium conditions are not always binding; indeed, divergence from the equilibrium path is frequently observed under realistic parameter configurations. The results suggest that adaptive learning processes like Q-learning can augment or diminish the presence of endogenous instability in actual relationships.

# 1 Introduction

This project is a simulation of the unstable relationships model devised by Burdett, Imai, and Wright [1], a simulation programmed with artificially intelligent (AI) learning agents. It is much in the spirit of Marimon, McGrattan, and Sargent [6] who adapted Kiyotaki and Wright's search model of monetary exchange and implemented a Holland-type classifier learning system to guide the trading and consumption decisions of AI agents. Along with the shift in application focus, I employ a different model of AI learning – Q-learning – in contrast to the classifier-systems approach of MMS.

In BIW, agents seek rewards by entering into bilateral relationships of indefinite length with randomly matched partners. Relationships are conceived in the abstract – they are meant to encompass a wide range of economic partnerships from job seeking to club formation to marital courtship. To rule out adverse selection effects, potential match partners are identical *ex ante* and rewards are relationship-specific, i.e. a partnership is an experience good. Agents can search for a different partner while matched which means that the rate of match dissolution is endogenous. Partners cannot restrict each other's search behaviour so a partnership has the form of a non-cooperative game with simultaneous moves. What is interesting about the BIW model is the notion of endogenous instability: agents might anticipate that partners will search while matched and discount the value of matches accordingly. This speculative discounting reinforces the incentive to search while matched and thus may influence the equilibrium outcome. Endogenous instability gives rise to multiple equilibria where relationships can be alternately stable (no search while matched) or unstable (search while matched) depending on beliefs.

As highlighted by MMS, one issue that is readily addressed by a simulation model with AI agents is equilibrium selection: when there exist multiple equilibria, how often, if ever, do agents coordinate on a particular equilibrium? MMS found that one 'fundamental' equilibrium dominated in their simulations despite the possibility of other equilibria. It remains to be seen whether this

result extends to BIW. In [4], I found BIW's assumption of non-transferrable utility to be crucial: if partners can commit to contracts that specify search-contingent side payments, Coasian arguments imply that endogenous instability is no longer present. This suggests to me that intelligent agents with the capacity to learn and perhaps with prior beliefs biased towards commitment might avoid the unstable equilibria in some instances.

This project, like MMS and other reinforcement learning (RL) applications in economics, is motivated by the bounded rationality paradigm promulgated by Sargent [7] among others. In conventional rational expectations (RE) models, the equilibrium laws of motion are assumed to be common knowledge and agents choose best responses accordingly: they have perfect foresight in the sense that they know that deviations from the equilibrium path should never occur. This contrasts sharply with RL models in which the equilibrium laws of motion are unknown and can only be learned from actual experience. Insofar as bounded rationality and learning pervades decision-making in the real world, simulation models with artificially intelligent agents can offer insight into the behaviour of naturally intelligent agents.

## 2 Formulating BIW as a Q-learning problem

The generic DP problem facing a single agent is to choose an action today to maximize expected utility – typically, a discounted sum of uncertain future rewards – over some horizon. Uncertainty is captured by a state variable that may depend on the history of agent actions and environmental responses. The decision environment is said to be Markovian if the current state and action are sufficient to determine the next state transition. In Watkin's Q-learning model [9], subjective evaluations known as Q-values are assigned to each state-action combination – the Q-values are supposed to represent the expected discounted utility of taking a particular action now and acting optimally thereafter, conditional on a particular state being realized. As rewards and state realizations occur, the Q-values are updated in a one-step-ahead manner. Formulating the

Q-learning problem in a dynamic progamming context amounts to a specification of the (observable) state space, the reward structure, the set of feasible actions or policies, and the state transition process(es).

For the problem at hand, I define the state as the current match reward in conjunction with the current match offer, which I denote as $(x, y)$. Time is measured in discrete periods; in each period, the agent observes $(x, y)$ and selects an action denoted $(a, s)$. An action is a pair of binary decisions: a decision to accept or reject the current match offer $(a = 1, 0)$ and a decision to search for new matches $(s = 1, 0)$. From the agent's perspective, a state transition occurs when a new partnership is formed, when the current partnership is dissolved, or when a new match offer is drawn. Since both exogeneous and partner-induced separations are tantamount to reward shocks, there is no need to include these events as additional state variables although it proves useful in the implementation to treat them as such. The match offer $y$ is a random variable with a fixed distribution. Offers are received only if the agent decides to search in the previous period; if search is unsuccessful, $y$ is assigned a null value. BIW assume that single, unmatched agents always engage in search and, for simplicity, I further assume that single agents receive a match reward of zero. Matched agents who search pay a fixed search cost; for single agents, the search cost is normalized to zero. In the simulation, I treat these search costs as non-positive match rewards.

If the information is perfect, the environment is clearly Markovian: prior knowledge of $(x, y, a, s)$ for every agent is sufficient to predict the next state transition. Moreover, the current match reward $x$ is sufficient to characterize the equilibrium of the system – BIW derive dynamic programming (DP) equations that pin down the value of matches conditional on the match reward.

In the case of just two match types, the DP equations are:

$$\begin{aligned} rV_0 &= \alpha\pi a_2 A_2(V_2 - V_0) + \alpha(1-\pi)a_1 A_1(V_1 - V_0) \\ rV_1 &= x_1 + (\sigma + S_1\alpha)(V_0 - V_1) + s_1\Sigma_1 \\ rV_2 &= x_2 + (\sigma + S_2\alpha)(V_0 - V_2) + s_2\Sigma_2 \end{aligned}$$

$$\begin{aligned} \Sigma_1 &= \alpha\pi[a_2 A_2 V_2 + (1 - a_2 A_2)V_0 - V_1] + \alpha(1-\pi)(1 - a_1 A_1)(V_0 - V_1) - \delta \\ \Sigma_2 &= \alpha\pi(1 - a_2 A_2)(V_0 - V_2) + \alpha(1-\pi)[a_1 A_1 V_1 + (1 - a_1 A_1)V_0 - V_2] - \delta \end{aligned}$$

The DP equations are interpreted as follows. The subscripts 0, 1, and 2 correspond to the states of being single, currently in type-1 match with reward $x_1$, and currently in a type-2 match with reward $x_2 > x_1$, respectively. $V_i$ denotes the net present value of being in state i and $r = \gamma^{-1} - 1$ denotes the time rate of preference so that $rV_i$ represents the net flow of rewards or 'interest income' accruing to the relationship 'asset'. An agent's decision to accept a type-i match is denoted $a_i$ and the decision to search while in a type-i match is denoted $s_i$ ; the mirror decisions of a representative partner are denoted $(A_i, S_i)$. While in the single state, accepting a match results in a 'capital gain' of $V_i - V_0$. While in a match state, exogenous separations occur with probability $\sigma$ and partner-induced separations occur with probability $S_i\alpha$ , i.e., the probability that one's partner accepts an outside offer. If an agent engages in search while matched, a new match offer is received with probability $\alpha$; the offer is a type-2 match with probability $\pi$ and a type-1 match with probability $1 - \pi$. The net benefit of search while in a type-i match is given by the expressions $\Sigma_i$ and the cost of search for both match types is a fixed amount $\delta$. (Recall that the search cost for singles is zero by assumption.) BIW assume a break-up rule that precludes returning to a previous match once a match offer is accepted. This means that receiving a type-j match offer while in a type-i match results in a capital loss of either $V_j - V_i$ or $V_0 - V_i$ depending on whether the offer

is accepted or rejected. The break-up rule simplifies the equilibrium analysis because it delivers an optimal policy that does not depend on the match state. (The rule can be relaxed in the simulation.)

In the case of just two match types, a Nash-Markov equilibrium in pure strategies specifies an action policy $(a_1, s_1; a_2, s_2)$ that maximizes the values expressed by the DP equations; the equilibrium is said to be symmetric if $(a_i, s_i) = (A_i, S_i)$ for both match types. The break-up rule means that the equilibrium policy is invariant over all match states. Ultimately, the equilibrium match values depend on three exogenous factors: the distribution of match offers, the cost of search, and the match technology, i.e. exogneous probabilities of meeting and separating.

We can now proceed to derive expressions for the Q-values based on the DP equations. Define $Q(x, y, a, s)$ as the value of taking action $(a, s)$ in the current state $(x, y)$ and acting optimally thereafter:

$$\begin{aligned} Q(x,y,a,s) &= \gamma a\{y + s(-\delta_y + E[\max_{a',s'} Q(x',y',a',s')|y, s=1]) \\ &\quad +(1-s)E[\max_{a',s'} Q(x',y',a',s')|y, s=0]\} \\ &+ \gamma(1-a)\{x + s(-\delta_x + E[\max_{a',s'} Q(x',y',a',s')|x, s=1] \\ &\quad +(1-s)E[\max_{a',s'} Q(x',y',a',s')|x, s=0]\} \end{aligned}$$

Conditional expectations are taken with respect to the match rewards, offers, and separations to be realized in the subsequent period. In the case of just two match types, expectations reflect the fact that match offers arrive at a rate of $\alpha$ per period, offers are type-2 with probability $\pi$, etc. The possiblity of separation means that next period's match reward $x'$ may differ from the current match reward $x$, even if the agent decides $s = 0$. After applying the expectation and maximum operators to both sides and substituting $\gamma = \frac{1}{1+r}$, it is a matter of algebra to show that the Q-value expression is consistent with the DP equations.

Rather than work with Q-values over the unrestricted state-action space, I

find it convenient to work with a bivariate system of Bellman equations in terms of match values $V(x, y, a)$ and search values $Q(z, s)$. The Bellman equations are:

$$\begin{aligned} V(x, y, a) &= [ay + (1 - a)x] + \max_s Q([ay + (1 - a)x], s) \\ Q(z, s) &= -\delta_z s + \gamma E[\max_{a'} V(x', y', a')|z, s] \end{aligned}$$

Here, the agent's optimal or greedy action is to accept if $V(x, y, 1) > V(x, y, 0)$ and, given $z = ay + (1 - a)x$, to search if $Q(z, 1) > Q(z, 0)$. Intuitively, agents should realize that accepting the match offer $y$ or rejecting it in favour of $x$ yields the selected match reward plus the expected value of search given the acceptance outcome $z$. By the same token, the decision to search this period yields the expected value of next period's match minus the search cost $\delta$ conditonal on $z$. These considerations are reflected in the Bellman equations.

The V- and Q-value estimates are updated according to the following rules:

$$\begin{aligned} V(z_t) &\leftarrow V(z_t) + \beta_t[z_t + \max_s Q(z_t, s) - V(z_t)] \\ Q(z_t, s_t) &\leftarrow Q(z_t, s_t) + \beta_t[-\delta s_t + \gamma \max_a V(z_{t+1}(a)) - Q(z_t, s_t)] \\ V_0 &\leftarrow V_0 + \beta_t[0 + Q_0 - V_0] \\ Q_0 &\leftarrow Q_0 + \beta_t[0 + \gamma \max\{V(y_{t+1}), V_0\} - Q_0] \end{aligned}$$

$$\begin{aligned} z_t &\equiv a_t y_t + (1 - a_t)x_t \\ z_{t+1}(a) &\equiv a y_{t+1} + (1 - a)x_{t+1} \end{aligned}$$

To wit, the Q-learning agent observes the state transition $(x_t, y_t) \rightarrow (x_{t+1}, y_{t+1})$ and then selects the V- and Q-value estimates to be updated given the previous action $(a_t, s_t)$. Notice that the expectation terms in the Bellman equations are replaced with their most recently observed counterparts. The learning rate $\beta_t$ regulates the degree to which the new estimate is dominated by the most recent

update; in the simulation I allow $\beta_t$ to decrease exponentially over time. As more of the state space is explored, the expectation is that subsequent updates bring the V- and Q-value estimates closer to an equilibrium solution.

# 3 Implementation details

Here I provide a high-level description of the simulation code. Specific code elements are discussed in the appendix. The logic of the Q-learning algorithm was hinted at in the previous section; for clarification purposes, the pseudo-code from Glorennec [2] appears below:

**Algorithm of Q($\lambda$)**

$t = 0$, $Q(x,a) = 0$, $e_t(x,a) = 0$ for all $(x,a)$ of $\mathcal{X} \times \mathcal{A}$
**observe** $x_t$
**repeat**
  **choose** $a_t$
  **observe the transition** $x_t \rightarrow x_{t+1}$
  $\epsilon_t^\dagger = r_t + \gamma V_t(x_{t+1}) - Q(x_t, a_t)$
  $\epsilon_t = r_t + \gamma V_t(x_{t+1}) - V_t(x_t)$
  **calculate** $e_t(x,a)$, equation 25 or 26
  **calculate** $Q(x,a)$ for all $(x,a)$ of $\mathcal{X} \times \mathcal{A}$:
    $Q(x_t, a_t) \leftarrow Q(x_t, a_t) + \beta \epsilon_t^\dagger e_t(x_t, a_t)$
    $Q(x,a) \leftarrow Q(x,a) + \beta \epsilon_t e_t(x,a)$ for all pairs different from $(x_t, a_t)$
  $t \leftarrow t + 1$
**until a stopping condition**

In the above, $(x_t, a_t)$ represents the state-action pair at date $t$, $r_t$ is the immediate reward, and $V_t(x) = \max_a Q(x,a)$; we can interpret this code in the present context using the state, action, and Q-value definitions from the previous section. The parameter $\lambda$ relates to a generalization of Q-learning, Q($\lambda$)-learning, that applies the current update to state-action pairs visited prior to $(x_t, a_t)$ weighted by the eligibility trace values $\epsilon_t$; the case of $\lambda = 0$ is equivalent to standard Q-learning. The implementation allows for an unrestricted $\lambda$ but I generally ignore this feature in the following discussion.

For the most part, the implementation adheres to the algorithm sketched

in the pseudo-code. Each trial in the simulation represents an interaction between the agent and the environment, the latter standing in for the agent's current or potential match partner. When the simulation starts, the agent's V- and Q-values are initialized to zero to reflect the agent's lack of experience. At the start of each trial, the environment processes the agent's previous acceptance and search decisions $(a_t, s_t)$ to realize the next state transition $(x_t, y_t) \rightarrow (x_{t+1}, y_{t+1})$. The agent subsequently observes the new match state, selects a new action $(a_{t+1}, s_{t+1})$, and updates the values $V(x_t, y_t, a_t)$ and $Q(x_t, y_t, s_t)$ in light of the new state information. This sequence is repeated until a predetermined stopping criterion is met.

The simulation code was written in the Java programming language. Kerr et al [5] developed the JavaRL code library specifically for RL applications; it supports a number of popular RL algorithms including Q($\lambda$)-learning. In the implementation, I maintain the basic JavaRL archictecture and specialize the core classes where necessary. The code library defines a class hierarchy that extends from the abstract to the specific; it also divides into class modules grouped by function. In keeping with object-oriented programming principles, higher-level interactions between modules are defined so as not to depend on lower-level representations internal to the module.

At the top of the JavaRL class hierarchy are three superclasses: Agent, Environment, and Simulation. The Agent class maps out the general data structures and operations that constitute the agent's decision-making apparatus; in the Q-learning context, these are chiefly the selection of an action given the current state and the invocation of Q-learning procedures in response to an action result. The Environment class defines the feedback channels that influence the agent's behaviour; implementing this class amounts to simulating the assumed state transition process. Finally, the Simulation class integrates the simulation tasks to be executed; it initializes the Agent and Environment objects, calls the various agent-interaction procedures at run-time, and reports the simulation results back to the user.

# 4 Results

The aim of the simulation exercise is to characterize the behaviour of Q-learning agents in the BIW environment and relate this behaviour to the RE equilibrium predictions. To motivate the experimental setup, I begin this section with a review of BIW's Proposition 1 and the mapping of RE equilibria to specific regions of the parameter space. I then discuss the parameter configurations and experimental results.

Proposition 1 of BIW establishes the necessary and sufficient conditions for a symmetric RE equilibrium in the case of just two match types. These equilibrium conditions pin down the optimal policy subject to inequality restrictions on the parameter space – the relevant parameters include $x_1, x_2, \delta, r, \alpha, \pi$, and $\sigma$. Recall that a policy $(a_1, s_1; a_2, s_2)$ specifies a pair of binary decisions for each match type. Proposition 1 says that out of a possible sixteen policies only five constitute a symmetric RE equilibrium given $x_2 \geq x_1 \geq 0$. One equilibrium, the "degenerate" case, is ruled out by the assumption that singles receive a match reward of zero. Table 1 lists the four non-degenerate equilibrium policies:

For fixed values of $\delta, r, \alpha, \pi$, and $\sigma$, each pair of equilibrium conditions defines a closed region in $(x_1, x_2)$-space. These regions are illustrated in the equilibrium map above. (The map is not drawn to scale.) Overlap across regions is indicative of multiple equilibria, e.g., in the triangular area bordered by $y_2, y_3$,

| $\text{Eq}^{\text{m}}$ type | $\text{Eq}^{\text{m}}$ policy | $\text{Eq}^{\text{m}}$ conditions |
|---|---|---|
| choosy (C) | $(R, -; A, N)$ | $x_1 \leq \delta$ and $x_2 \geq y_1$ |
| faithful (F) | $(A, N; A, N)$ | $x_2 \leq y_1$ and $x_2 \leq y_2$ |
| unfaithful (U) | $(A, S; A, N)$ | $x_1 \geq \delta$ and $x_2 \geq y_3$ |
| perverse (P) | $(A, N; A, S)$ | $x_2 \geq \delta$ and $x_1 \geq y_4$ |
| | | $y_1 \equiv \frac{r+\sigma+\alpha\pi}{\alpha\pi} x_1$ |
| | | $y_2 \equiv x_1 + \frac{r+\sigma}{\alpha\pi}\delta$ |
| | | $y_3 \equiv x_1 + \frac{r+\sigma}{\alpha\pi}\delta + \frac{\alpha}{r+\sigma+2\alpha}(\delta - x_1)$ |
| | | $y_4 \equiv x_2 + \frac{r+\sigma}{\alpha(1-\pi)}\delta + \frac{\alpha}{r+\sigma+2\alpha}(\delta - x_2)$ |

Table 1: The four non-degenerate equilibria

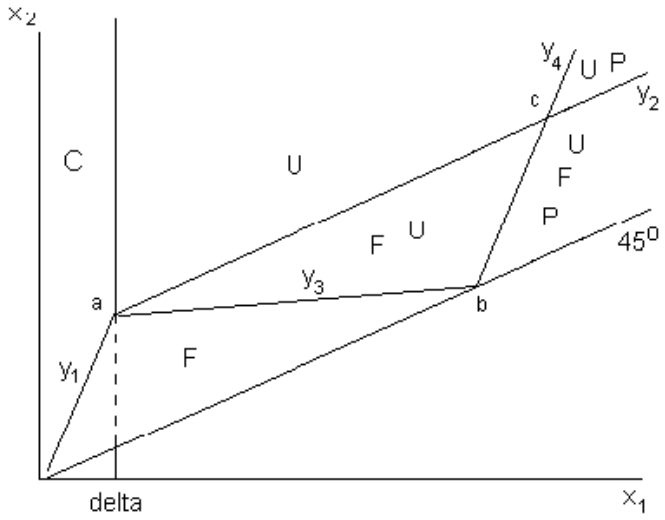


Figure 1: The equilibrium map

and $y_4$ both F- and U-type equilibria are possible. Where there is overlap one would expect the equilibrium outcome to depend on the formation of agent beliefs, given rational expectations; elsewhere one would expect the outcome to be independent of beliefs.

To the extent that beliefs are manifest in the learning process via the Q-value updates, we can gauge how beliefs influence equilibrium selection from the distribution of observed policy outcomes. We might, for instance, configure the parameters so that only the F-type equilibrium is supported but nonetheless observe U-type policies or other non-equilibrium outcomes; this might then lead us to conclude that the equilibrium predictions are an inadequate characterization of agent behaviour. I will come back to this point in the ensuing discussion.

A thorough simulation analysis would call for a full slate of comparative-statics experiments spanning the entire parameter space; this is left for future study. Here, the scope is more modest with a focus on discerning qualitative differences, if any, between the simulated and predicted outcomes. Each experiment consists of six simulation runs taking one $(x_1, x_2)$ point from each of the equilibrium regions in Figure 1; the same points are used in all of the experiments. I also consider the cusp points corresponding to points $a$, $b$, and $c$ on the equilibrium map. The base parameter configuration is $\delta = 1, r = .05, \alpha = \pi = .5, \sigma = .2$ which fixes the cusp point coordinates at $(1, 2), (3.5, 3.5), (5, 6)$.[1] Each experiment shifts one of the parameters away from the base configuration – I restrict attention to shifts in $\delta, \alpha$ and $\sigma$. In general, parametric shifts will affect the shape and location of the equilibrium regions and hence the predicted outcomes. Owing to such a shift we might find, for instance, what used to be a cusp point now lies inside one of the new equilibrium regions. Deviations from the predicted equilibrium dynamics would be further indication of a behavioural bias due to Q-learning.

[1] Given $\pi = .5$, the cusp point coordinates are given by $a = (\delta, (1 + \mu_0)\delta)$, $b = ((1 + \mu_1)\delta, (1 + \mu_1)\delta)$,$c = ((1 - \mu_0 + 2\mu_1)\delta, (2 - \mu_0 + 2\mu_1))$ where $\mu_0 \equiv \frac{r+\sigma}{\alpha\pi}$ and $\mu_1 \equiv \mu_0 / \frac{\alpha}{r+\sigma+2\alpha}$. In the base configuration we have $\delta = \mu_0 = 1, \mu_1 = 2.5$; applying the formulae yields the given coordinates.

### 4.1 Results for the base configuration

The basic setup for the simulation experiments is the following. A simulation run consists of a series of trials wherein the Q-learning agent interacts with the environment over the course of a fixed number of steps or periods. We can think of the trial series as a random sample of agents from the population-at-large. At the start of each trial, the Q-values are set to 0.0 and the learning rate to 1.0 so as to restart the Q-learning process anew. The Q-values are updated and the learning rate decays as the trial continues. At the end of the trial, the last policy chosen by the agent – the cumulative result of the Q-learning process – is recorded. After all of the trials are concluded, a variety of statistics are reported including a breakdown of the observed policy distribution.

In Table 2, I report simulation results for the base parameter configuration: $\delta = 1, r = .05, \alpha = \pi = .5, \sigma = .2$. I fix the trial length at N=300 and associate trial steps with months so that 10 steps corresponds to roughly a year: a time frame of 30 years is conceivable in the context of career or marriage search. I also fix the probability of random action $\epsilon = .1$ and the lower bound for the learning rate $\beta = .01$.[2] In this environment, the average time spent seeking a match is two months, type-1 and type-2 match offers are equally likely, and stable matches endure for half a year on average. A series of 1000 trials was executed for selected values of $(x_1, x_2)$. In all trials, the pre-experimental stage option was skipped and the partner behaviour option optPlay was set to true; also, BIW's break-up rule applies throughout.

The observed policy distributions in Table 2 are clearly non-degenerate: deviations from the equilibrium predictions do occur. Note in particular the frequency of outcomes in the 'Other' category not associated with any RE equilibrium; in many of those cases there is search while in a type-2 match or even outright rejection of type-2 matches. These findings appear to be inconsistent with convergence to equilibrium. However, it may be that the agent's capacity

[2] The fixed lower bound for $\beta$ means that the lifespan-adjusted rate of decay of $\beta$ is constant. Alternatively, one could allow the lower bound for $\beta$ to vary with N but this would conflate the effects of $\beta$ and N on the simulation results, i.e., the shape of the learning curve vs. the number of agent-environment interactions.

| $(x_1, x_2)$ | Predicted eq$^{\text{m}}$ | C | F | U | P | Other | $\Sigma$ |
|---|---|---|---|---|---|---|---|
| 0.5, 2.0 | C | 557 | 205 | 0 | 19 | 219 | 1000 |
| 1.5, 2.0 | F | 292 | 295 | 3 | 88 | 322 | 1000 |
| 2.5, 3.0 | F,U | 258 | 158 | 27 | 103 | 454 | 1000 |
| 3.5, 5.0 | U | 255 | 98 | 13 | 136 | 498 | 1000 |
| 5.5, 6.0 | F,U,P | 194 | 88 | 57 | 92 | 569 | 1000 |
| 6.5, 8.0 | U,P | 193 | 81 | 36 | 108 | 582 | 1000 |

Table 2: Base case simulation with trial length N=300

| $(x_1, x_2)$ | Predicted eq$^{\text{m}}$ | C | F | U | P | Other | $\Sigma$ |
|---|---|---|---|---|---|---|---|
| 0.5, 2.0 | C | 65 | 1 | 0 | 2 | 32 | 100 |
| 1.5, 2.0 | F | 43 | 0 | 0 | 0 | 57 | 100 |
| 2.5, 3.0 | F,U | 27 | 5 | 0 | 0 | 68 | 100 |
| 3.5, 5.0 | U | 27 | 0 | 0 | 0 | 73 | 100 |
| 5.5, 6.0 | F,U,P | 29 | 1 | 0 | 0 | 70 | 100 |
| 6.5, 8.0 | U,P | 35 | 0 | 0 | 0 | 65 | 100 |

Table 3: Base case simulation with trial length N=3000

to learn from experience is limited by the number of trial steps taken. Convergence of the Q-learning algorithm might be slow requiring many steps to approach the equilibrium path.

To test whether the findings in Table 2 are robust, I ran the base simulation again with trial lengths N=3000 and N=30000; because of the increased computational burden, I also reduced the sample size to $\Sigma = 100$. The results are presented in Tables 3 and 4. When the trial length is extended to N=3000, policies in the 'Other' category are more prevalent and, of the four equilibrium policies, only the C-type is observed in significant numbers. This is a surprising result if one is expecting convergence to the equilibrium solution. When the trial length is extended to N=30000, however, the picture is different. Table 4 shows closer agreement with the equilibrium predictions: in every case a clear majority of the trials result in the predicted equilibrium outcome.

How should one interpret these results? On the face of it, the fact that N=30000 fits the equilibrium predictions better than N=300 or N=3000 lends

| $(x_1, x_2)$ | Predicted eq$^{\text{m}}$ | C | F | U | P | Other | $\Sigma$ |
|---|---|---|---|---|---|---|---|
| 0.5, 2.0 | C | 98 | 0 | 2 | 0 | 0 | 100 |
| 1.5, 2.0 | F | 0 | 72 | 21 | 7 | 0 | 100 |
| 2.5, 3.0 | F,U | 0 | 75 | 22 | 3 | 0 | 100 |
| 3.5, 5.0 | U | 0 | 12 | 82 | 6 | 0 | 100 |
| 5.5, 6.0 | F,U,P | 0 | 59 | 25 | 16 | 0 | 100 |
| 6.5, 8.0 | U,P | 0 | 21 | 72 | 7 | 0 | 100 |

Table 4: Base case simulation with trial length N=30000

support to the convergence hypothesis. That is, given sufficient capacity to learn from experience, Q-learning behaviour tends toward RE equilibrium and any non-equilibrium behaviour is due to insufficient experience. The fact that the 'Other' category of Table 4 is a column of zeros also points to convergence. On the other hand, the trial length should reflect the time scale at which relationship events are experienced in reality. If a relationship history of length N=300 is deemed appropriate given the domain of choice, how can histories 100 times as long be sensible? One could argue that the learning model assumes too great of an information gap. Along with learning by trial-and-error, people learn and form beliefs about relationship environments in other ways, e.g. via social networks, the popular media, cultural influences, etc. Extending the trial length to N=30000 "makes up" for information resources that would, if they were available to the agent, drive the learning process to convergence. In this interpretation, the results for N=30000 suggest that the RE model is a good predictor of learning outcomes insofar as one rejects the notion that relationships are a pure experience good. Conversely, if one insists that relationships are mostly an experience good and that realism demands N=300, the predictive power of rational expectations appears to be weak. Which is the better interpretation obviously depends on the specific application.

Concerns about realism aside, I restrict N=30000 in the following experiments to precisely understand the effects of parametric shifts. Starting from the base configuration, I proceed to look at shifts in the search cost $\delta$, the match offer frequency $\alpha$, and the probability of exogenous separation $\sigma$.

| $(x_1, x_2)$ | Predicted eq$^{\text{m}}$ | C | F | U | P | Other | $\Sigma$ |
|---|---|---|---|---|---|---|---|
| 0.5, 2.0 | U | 21 | 0 | 79 | 0 | 0 | 100 |
| 1.5, 2.0 | U | 0 | 2 | 91 | 4 | 0 | 100 |
| 2.5, 3.0 | U | 0 | 4 | 86 | 7 | 3 | 100 |
| 3.5, 5.0 | U | 0 | 3 | 96 | 1 | 0 | 100 |
| 5.5, 6.0 | U | 0 | 12 | 65 | 20 | 3 | 100 |
| 6.5, 8.0 | U | 0 | 6 | 85 | 5 | 4 | 100 |

Table 5: Experiment with $\delta = 0$ and N=30000

| $(x_1, x_2)$ | Predicted eq$^{\text{m}}$ | C | F | U | P | Other | $\Sigma$ |
|---|---|---|---|---|---|---|---|
| 0.5, 2.0 | C | 99 | 0 | 1 | 0 | 0 | 100 |
| 1.5, 2.0 | F | 18 | 76 | 2 | 1 | 3 | 100 |
| 2.5, 3.0 | F | 15 | 78 | 11 | 7 | 0 | 100 |
| 3.5, 5.0 | F,U | 3 | 61 | 35 | 1 | 0 | 100 |
| 5.5, 6.0 | F | 0 | 69 | 25 | 6 | 0 | 100 |
| 6.5, 8.0 | F,U | 0 | 55 | 34 | 11 | 0 | 100 |

Table 6: Experiment with $\delta = 2$ and N=30000

## 4.2 Varying the cost of search

Two different values for the search cost are considered: $\delta = 0$ and $\delta = 2$. In the case of $\delta = 0$, all of the areas associated with the C- and F-type equilibria in Figure 1 disappear leaving the U-type equilibrium as the unique predicted outcome. In the case of $\delta = 2$, the predicted change is less dramatic: the equilibrium regions are larger relative to the base case but the equilibrium map is essentially the same as Figure 1.[3] The results are reported in Tables 5 and 6. For both $\delta = 0$ and $\delta = 2$, there is ample agreement with the equilibrium predictions; in most cases, a strong majority of the trials result in the predicted U-type equilibrium.

[3]The cusp point coordinates for $\delta = 2$ are $(2, 2), (7, 7), (10, 12)$, scaling the original cusp point coordinates by a factor of 2. The slopes of the borders partitioning the regions are the same as before.

| $(x_1, x_2)$ | Predicted eq$^{\mathrm{m}}$ | C | F | U | P | Other | $\Sigma$ |
|---|---|---|---|---|---|---|---|
| 0.5, 2.0 | C | 99 | 1 | 0 | 0 | 0 | 100 |
| 1.5, 2.0 | F | 1 | 86 | 9 | 4 | 0 | 100 |
| 2.5, 3.0 | F | 0 | 88 | 9 | 3 | 0 | 100 |
| 3.5, 5.0 | U | 0 | 84 | 15 | 1 | 0 | 100 |
| 5.5, 6.0 | F,U | 0 | 78 | 16 | 6 | 0 | 100 |
| 6.5, 8.0 | U | 0 | 66 | 26 | 8 | 0 | 100 |

Table 7: Experiment with $\alpha = .3$ and N=30000

| $(x_1, x_2)$ | Predicted eq$^{\mathrm{m}}$ | C | F | U | P | Other | $\Sigma$ |
|---|---|---|---|---|---|---|---|
| 0.5, 2.0 | C | 100 | 0 | 0 | 0 | 0 | 100 |
| 1.5, 2.0 | F,U | 2 | 56 | 34 | 8 | 0 | 100 |
| 2.5, 3.0 | F,U | 2 | 43 | 39 | 16 | 0 | 100 |
| 3.5, 5.0 | U | 7 | 14 | 70 | 9 | 0 | 100 |
| 5.5, 6.0 | F,U,P | 0 | 19 | 61 | 20 | 0 | 100 |
| 6.5, 8.0 | U,P | 0 | 6 | 72 | 22 | 0 | 100 |

Table 8: Experiment with $\alpha = .7$ and N=30000

### 4.3 Varying the frequency of match offers

Two different values for the match offer frequency are considered: $\alpha = .3$ and $\alpha = .7$. An increase in $\alpha$ has the effect of depressing the areas associated with the F-type equilibrium in Figure 1 relative to the other equilibrium types. This can be seen directly in the expression for $y_1$ wherein an increase in $\alpha$ shifts the $x_2$- intercept downward; the same can be said for $y_2$.[4] Intuitively, a higher frequency of match offers lowers the marginal value of being matched. This gives agents greater incentives to search while matched and to reject offers outright. The results are reported in Tables 7 and 8: again, there is ample agreement with equilibrium predictions except for the case of $(x_1, x_2) = (6.5, 8.0)$ with $\alpha = .3$. It should be noted, however, that in the exceptional case, the point $(x_1, x_2) = (6.5, 8.0)$ is near the border separating the F,U region from the U region in Figure 1.

[4]For $\alpha = .3$, the cusp point coordinates are $(1.0, 2.7), (5.7, 5.7), (8.8, 9.8)$ approximately. For $\alpha = .7$, the cusp point coordinates are $(1.0, 1.7), (2.7, 2.7), (3.7, 4.7)$ approximately.

| $(x_1, x_2)$ | Predicted eq$^{\text{m}}$ | C | F | U | P | Other | $\Sigma$ |
|---|---|---|---|---|---|---|---|
| 0.5, 2.0 | C | 98 | 0 | 0 | 2 | 0 | 100 |
| 1.5, 2.0 | F,U | 18 | 36 | 33 | 12 | 1 | 100 |
| 2.5, 3.0 | F,U | 10 | 22 | 34 | 33 | 1 | 100 |
| 3.5, 5.0 | U | 8 | 4 | 58 | 29 | 1 | 100 |
| 5.5, 6.0 | F,U,P | 2 | 16 | 38 | 43 | 1 | 100 |
| 6.5, 8.0 | F,U,P | 3 | 3 | 65 | 28 | 1 | 100 |

Table 9: Experiment with $\sigma = .1$ and N=30000

### 4.4 Varying the probability of exogenous separation

Two different values for the probability of exogenous separation are considered: $\sigma = .1$ and $\sigma = .3$. An increase in $\sigma$ has the opposite effect to that of an increase in $\alpha$: it enlarges the areas associated with the F-type equilibrium at the expense of the other equilibrium types.[5] The intuition is analogous to varying the match offer frequency: a higher probability of separation increases the marginal value of being matched which reduces the incentives to reject match offers and to search while matched. Indeed, the equilibrium predictions are almost exactly parallel to those in Tables 7 and 8. The results are reported in Tables 9 and 10. For $\sigma = .1$, agreement with the equilibrium predictions is mixed; the incidence of P-type equilibrium outcomes is greater even in those cases where it does not constitute an equilibrium. For $\sigma = .3$, there is tighter agreement with the equilibrium predictions with fewer P-type equilibrium outcomes and the elimination of 'Other' outcomes entirely.

### 4.5 Introducing two-stage learning

In the previous experiments, an agent selects the random action with probability $\epsilon = .1$ and the current optimal action with probability $1 - \epsilon$ over the course of a simulation trial. In this experiment, the selection of random and optimal actions is separated into two stages. During the first stage of length $N_1 = N\epsilon$ the agent always selects the random action while during the second stage of

[5] For $\sigma = .1$, the cusp point coordinates are $(1.0, 1.6), (2.4, 2.4), (3.2, 4.2)$ approximately; for $\sigma = .3$, the cusp point coordinates are $(1.0, 2.4), (4.8, 4.8), (7.2, 8.2)$ approximately.

| $(x_1, x_2)$ | Predicted eq$^m$ | C | F | U | P | Other | $\Sigma$ |
|---|---|---|---|---|---|---|---|
| 0.5, 2.0 | C | 100 | 0 | 0 | 0 | 0 | 100 |
| 1.5, 2.0 | F | 0 | 93 | 7 | 0 | 0 | 100 |
| 2.5, 3.0 | F | 0 | 92 | 6 | 2 | 0 | 100 |
| 3.5, 5.0 | U | 0 | 33 | 66 | 1 | 0 | 100 |
| 5.5, 6.0 | F,U | 0 | 84 | 8 | 8 | 0 | 100 |
| 6.5, 8.0 | U | 0 | 52 | 45 | 3 | 0 | 100 |

Table 10: Experiment with $\sigma = .3$ and N=30000

length $N_2 = N(1-\epsilon)$ the agent always selects the optimal action. This follows the approach of Hanaki et al [3] whose trials also consist of pre-experimental and experimental stages.

Here, I report the results of including a pre-experimental stage in the previous experiments. As before, I fix N=30000 and $\epsilon = .1$ which fixes the stage lengths $N_1 = 3000$ and $N_2 = 27000$. The results for the base configuration, $\delta = 2$, $\alpha = .7$, and $\sigma = .1$ are reported in Tables 11- 14. With each of the parameter configurations there is strong agreement with the equilibrium predictions, stronger than the original results without the pre-experimental stage. In the cases with multiple equilibria, one equilibrium type dominates; where F-,U-, and P-type equilibria are predicted, for instance, the U-type equilibrium outcome is observed most often. Outcomes in the 'Other' category are never observed; the same is true of the P-type equilibrium. As Hanaki et al [3] observed, including an initial phase of purely exploratory behaviour endogenizes the agents' prior beliefs so as to narrow the range of final outcomes.[6]

[6]Hanaki et al [3] found convergence to the efficient or welfare-maximizing outcome in a class of pure coordination games. Elimination of the P-type equilibrium and non-equilibrium outcomes is consistent with this result: Proposition 3 of BIW says that the efficient outcome (assuming a particular social welfare metric) is either a C-, F-, or U-type equilibrium. The prospect of endogenous instability in the BIW environment introduces an implicit coordination game between match partners.

| $(x_1, x_2)$ | Predicted eq$^{\text{m}}$ | C | F | U | P | Other | $\Sigma$ |
|---|---|---|---|---|---|---|---|
| 0.5, 2.0 | C | 100 | 0 | 0 | 0 | 0 | 100 |
| 1.5, 2.0 | F | 0 | 97 | 3 | 0 | 0 | 100 |
| 2.5, 3.0 | F,U | 0 | 72 | 28 | 0 | 0 | 100 |
| 3.5, 5.0 | U | 0 | 0 | 100 | 0 | 0 | 100 |
| 5.5, 6.0 | F,U,P | 0 | 5 | 95 | 0 | 0 | 100 |
| 6.5, 8.0 | U,P | 0 | 0 | 100 | 0 | 0 | 100 |

Table 11: Base simulation with $N_1 = 3000, N_2 = 27000$

| $(x_1, x_2)$ | Predicted eq$^{\text{m}}$ | C | F | U | P | Other | $\Sigma$ |
|---|---|---|---|---|---|---|---|
| 0.5, 2.0 | C | 100 | 0 | 0 | 0 | 0 | 100 |
| 1.5, 2.0 | F | 0 | 100 | 0 | 0 | 0 | 100 |
| 2.5, 3.0 | F | 0 | 100 | 0 | 0 | 0 | 100 |
| 3.5, 5.0 | F,U | 0 | 23 | 77 | 0 | 0 | 100 |
| 5.5, 6.0 | F | 0 | 100 | 0 | 0 | 0 | 100 |
| 6.5, 8.0 | F,U | 0 | 5 | 95 | 0 | 0 | 100 |

Table 12: Experiment with $\delta = 2, N_1 = 3000, N_2 = 27000$

| $(x_1, x_2)$ | Predicted eq$^{\text{m}}$ | C | F | U | P | Other | $\Sigma$ |
|---|---|---|---|---|---|---|---|
| 0.5, 2.0 | C | 100 | 0 | 0 | 0 | 0 | 100 |
| 1.5, 2.0 | F,U | 0 | 11 | 89 | 0 | 0 | 100 |
| 2.5, 3.0 | F,U | 0 | 0 | 100 | 0 | 0 | 100 |
| 3.5, 5.0 | U | 0 | 0 | 100 | 0 | 0 | 100 |
| 5.5, 6.0 | F,U,P | 0 | 0 | 100 | 0 | 0 | 100 |
| 6.5, 8.0 | U,P | 0 | 0 | 100 | 0 | 0 | 100 |

Table 13: Experiment with $\alpha = .7, N_1 = 3000, N_2 = 27000$

| $(x_1, x_2)$ | Predicted eq$^{\text{m}}$ | C | F | U | P | Other | $\Sigma$ |
|---|---|---|---|---|---|---|---|
| 0.5, 2.0 | C | 99 | 0 | 1 | 0 | 0 | 100 |
| 1.5, 2.0 | F,U | 0 | 20 | 80 | 0 | 0 | 100 |
| 2.5, 3.0 | F,U | 0 | 2 | 98 | 0 | 0 | 100 |
| 3.5, 5.0 | U | 0 | 0 | 100 | 0 | 0 | 100 |
| 5.5, 6.0 | F,U,P | 0 | 0 | 100 | 0 | 0 | 100 |
| 6.5, 8.0 | F,U,P | 0 | 0 | 100 | 0 | 0 | 100 |

Table 14: Experiment with $\sigma = .1, N_1 = 3000, N_2 = 27000$.

# 5 Conclusion

My preliminary investigation of Q-learning as applied to the unstable relationships model of Burdett, Imai and Wright has two main findings. First, the simulation results suggest that for selected parameter configurations the necessary conditions for convergence of Q-learning to RE equilibrium are strong: agents require long relationship histories to obtain the predicted equilibrium outcomes. Second, introducing two-stage learning as in Hanaki et al [3] exerts a significant push towards convergence.

Many extensions to the simulation model are possible. It would be interesting to see if the results are robust when there are more than two match types or even a continuous state representation, e.g. the fuzzy Q-learning extension discussed in Glorrenec [2]. The results may be sensitive to the infinite-horizon assumption as well. Todd, Billari, and Simao [8] discuss a variety of heuristic approaches to modelling and simulating marriage search behaviour with age-cohort considerations; exploring these ideas in the BIW context promises future research.

# APPENDIX

This appendix dicsusses the programming details of the implementation. In the following, I focus on the application-specific extension classes; the reader is referred to Kerr et al [5] for more details about the general JavaRL framework.

## Data structures

The JavaRL architecture has three basic data classes: State, Action, and ActionResult. Both State and Action are interfaces to be implemented by the user while ActionResult is a wrapper class for State and Action. The MatchState and MatchAction classes implement the State and Action interfaces. Instances of MatchState consist of a double value that represents the current match reward plus an instance of MatchEvent, a wrapper object consisting of a double value

representing the next match offer and a boolean value indicating whether an exogenous or partner-induced separation has occurred. MatchAction is yet another wrapper object with boolean values representing the agent's search and acceptance decisions.

Other special data classes include the interfaces MatchDist and QSurface together with their implementations DiscreteMathDist and TabQSurf. MatchDist is a facility for making random draws associated with the match offer and exogenous separation variables; DiscreteMathDist is a specific implementation of MatchDist that assumes a discrete support for the match offers. DiscreteMathDist also contains utilities for tracking the frequency of actions and states visited as well as a sampling procedure used by the MatchEnv class (described below) to select a representative partner at each simulation step. QSurface is an abstract interface used by the QLMatchAgent class to represent the Q-values and eligibility trace values; TabQSurf is a specific implementation that assumes a discrete, look-up table representation. A continuous state representation would require different implementations of MatchDist and QSurface.

### Extending the Agent class

The TDAgent class from the JavaRL library is an abstract subclass of Agent that supports temporal-difference (TD-) learning algorithms. Q-learning is a special case of TD-learning so I base the implementation on TDAgent rather than its Agent superclass.

The QLMatchAgent class extends TDAgent to implement Q-learning in the BIW context, making use of the specialized data structures MatchState, MatchAction, and MatchEvent throughout. QSurface objects are declared that provide access to the agent's Q-values and eligibility traces. The main method implemented by QLMatchAgent is step(), a method that returns the agent's next action – this overrides the definition in TDAgent. Supplemental methods fall into two categories: those used to select actions based on the current Q-values and those used to update the Q-values in response to new state information. In

the former category, getOptimalAction() and getRandomAction() implement the 'greedy' and random actions, respectively. In the latter category, getTDErrorForV() and getTDErrorForQ() compute the TD errors associated with the match and search values, respectively. My implementation of QLMatchAgent abstracts from whether the state has a discrete or continuous representation, provided the Q-values are accessible via the QSurface interface.

The TabQLMatchAgent class extends QLMatchAgent to define the Q-values and eligibility traces as look-up tables; it also implements a specialized update method that relies on this tabular representation. The init method instantiates the Q-value and eligibility trace objects as instances of TabQSurf. The updateEst method takes the TD error computed by getTDErrorForV() or getTDErrorForQ() and applies an update to the V- or Q- values for each state-action combination according to the Q($\lambda$)-learning algorithm; in the cae of $\lambda = 0$ only the most recent state-action is selected for updating because all other state-action combinations have a trace value of zero. Auxiliary methods provide iterator objects used by updateEst() to cycle through all of the state-action combinations. It is important to note that implementing a continuous state representation would require a different updating scheme altogether.

### Extending the Environment class

The MatchEnv class adds features to Environment that are specific to the BIW model. The step method is supplemented with methods that compute the state transitions and match rewards. Given the agent's current match state and action, a call to stateTransition() returns the agent's future match state. The stateTransition method, in turn, calls nextMatchEvent() to obtain a MatchEvent instance representing the agent's future match offer and separation status. The nextMatchEvent method calls on an instance of MatchDist to make a random draw from the match offer distribution if the agent previously chose to search. It also determines whether a separation occurs based in part on the actions of a representative partner realized by the auxiliary method partnerAccepts(),

e.g., a partner who accepts an outside match offer necessarily ends the current match. The simulation allows for various kinds of partner behaviour – these are described in the next subsection. Once the next match event is realized, the future match reward is adjusted to reflect the agent's future separation status and acceptance decision.

### Extending the Simulation class

The MatchSim class extends Simulation to incorporate application-specific features. It provides the command-line method main() that instantiates and initializes the actor objects – QLMatchAgent, MatchEnv, and MatchSim – and calls the trials method which initiates the simulation run. The learning rate $\beta_t$ is updated after every simulation step $t$; the auxiliary method calcBeta() returns a double value between 0.0 and 1.0 that decays with $t$ exponentially. Parameter values and initial conditions are set by main() and init(); initializations from init() feed forward to initializations of the agent and environment objects.

MatchSim makes it possible to run the simulation in two stages, first with the agent always choosing actions at random for a predetermined number of steps and afterwards choosing the optimal or greedy action with probability $1 - \epsilon$. This option creates an initial 'pre-experimental' stage that endogenizes the prior beliefs as in the experimental setup of Hanaki et al [3].

MatchSim also provides a utility that commits the representative partner to a predetermined action policy when the simPlay variable is set to true. This utility can be used to test if a symmetric RE equilibrium is stable in the sense of inducing the agent to choose the equilibrium policy when it is adopted by the representative partner. Calls to setPartnerAction() are necessary to fix the acceptance and search decisions for each match reward-offer combination. When simPlay is set to false, other options for controlling partner behaviour are available: setting randPlay to true forces the partner to always choose actions at random while setting optPlay to true forces the partner to always choose the optimal or greedy action. The default is for the partner to exactly mimic the

agent's behaviour according to the probability of selecting the random action vs. the optimal action, $\epsilon$.

# References


[1] Burdett, K., R Imai, and R Wright (2004), "Unstable relationships," *Frontiers of Macroeconomics*, **1**, 1-38.

[2] Glorennec, P.Y. (2000) "Reinforcement learning: an overview," *European Symposium on Intelligent Techniques*, Aachen, Germany.

[3] Hanaki, N., R Sethi, I Erev, A Peterhansl (2003), "Learning strategies," mimeo.

[4] Kang, M. (2002) "Paying for ties that bind: do transfer payments improve unstable relationships?" B.Sc. Honours thesis, University of Victoria.

[5] Kerr, A.J., TW Neller, CJ La Pilla, MD Schompert (2003), *Java Resources for Teaching Reinforcement Learning*, mimeo., Gettysburg College, PA.

[6] Marimon, R., E McGrattan, TJ Sargent (1990), "Money as a medium of exchange in an economy with artificially intelligent agents," *Journal of Economic Dynamics and Control*, **14**, 329-373.

[7] Sargent, T.J. (1993) *Bounded Rationality in Macroeconomics,* Clarendon Press, Oxford.

[8] Todd, P.M., FC Billari, J Simao (2002), "Age-at-marriage patterns can emerge from individual mate-search heuristics," mimeo.

[9] Watkins, C.J.C.H. (1989) *Learning from Delayed Rewards,* Ph.D. thesis, Cambridge University, UK.