THE
ECONOMIC
JOURNAL

The Economic Journal, **123** (*March*), 38–76. Doi: 10.1111/j.1468-0297.2012.02525.x. © 2012 The Author(s). The Economic Journal © 2012 Royal Economic Society. Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

SOCIAL LEARNING AND MONETARY POLICY RULES*

Jasmina Arifovic, James Bullard† *and Olena Kostyshyna*

We analyse the effects of social learning in a monetary policy context. Social learning might be viewed as more descriptive of actual learning behaviour in complex market economies. In our model, Taylor Principle governs uniqueness and expectational stability of rational expectations equilibrium (REE) under homogeneous recursive algorithms. We find that the Taylor Principle is not necessary for convergence to REE minimum state variable (MSV) equilibrium under social learning. Sunspot equilibria exist in the indeterminate region. Our agents cannot co-ordinate on a sunspot equilibrium in general form specification, however, they can co-ordinate on common factor specification. We contribute to the use of genetic algorithm learning in stochastic environments.

Recent research has emphasised how policy choices may influence the stability properties of rational expectations equilibrium. In a typical analysis, a policy maker may commit to a particular policy rule, stating how adjustments to a control variable will be made in response to disturbances to the economy. The policy rule together with optimising private sector behaviour may imply that there is a unique rational expectations equilibrium associated with the policy rule and that the equilibrium has desirable welfare properties. However, the equilibrium may or may not be robust to small expectational errors. If the expectations of the players in the economy are initially not rational but deviate from rational expectations by a small amount, behaviour of the players in the economy will be changed. This will then have effects on the price and quantity outcomes in the economy, feeding back into the learning process. Such a dynamic may or may not converge to the rational expectations equilibrium which is the policy maker's target. When the process does converge, it is called an *expectationally stable* or *learnable* equilibrium.

We study learnability in a standard context, the model of monetary policy of Woodford (2003). A standard result, discussed in Woodford (2003) and Bullard and Mitra (2002), is that in a simple version of the model, the rational expectations equilibrium will be learnable provided the policy maker follows the *Taylor Principle*.[1] This means that the policy maker must react sufficiently aggressively to economic developments, such as deviations of inflation from target or the deviation of output from the flexible price or potential level of output. Failure to do so will create a rational expectations equilibrium which is unstable in the recursive learning dynamic. Such an equilibrium is unlikely to be successfully implemented in actual policy making. Even small expectational errors would drive the economy away from the intended equilibrium. In addition, when the

* Corresponding author: Jasmina Arifovic, Department of Economics, Simon Fraser University, 8888 University Drive, Burnaby, BC, V5A 1S6, Canada. Email: arifovic@sfu.ca.

We thank Bruce McGough for helpful discussion and we thank two anonymous referees for helpful comments.

† Any views expressed are those of the authors and do not necessarily reflect the views of the Federal Reserve Bank of St. Louis or the Federal Reserve System.

[1] For a discussion of the Taylor Principle, Woodford (2001).

Taylor Principle is violated, equilibrium is indeterminate and sunspot equilibria exist in a neighbourhood of an indeterminate rational expectations equilibrium. In the recursive learning literature, it has generally been difficult to obtain expectational stability of sunspot equilibria.[2]

The standard results are derived under the assumption of homogeneous expectations which are updated via recursive algorithms. This is the approach discussed extensively in Evans and Honkapohja (2001). By assuming homogeneous expectations and recursive algorithms, analytical results can be obtained concerning the expectational stability properties of equilibria across a wide variety of models. In this article we study an alternative approach to learning, one that can be viewed as more realistic in terms of actual learning in complicated market economies. In it, agents are initially heterogeneous with respect to the models they use to forecast the future. Forecast rules are updated via genetic operators, meant to simulate the process of learning from neighbours and others in the economy. Results are not analytic but are based on computational experiments. We will call this alternative approach *social learning*.

Social learning has been studied in a wide variety of contexts in economics but not in the standard New Keynesian model where many of the other findings concerning learnability have been presented. One reason is that the New Keynesian model is inherently stochastic and the genetic algorithm applications which are drawn from the artificial intelligence literature are deterministic.[3] The genetic algorithm is meant to find 'good' solutions to complicated problems with no known best solution. One purpose of this article is to understand how insights from the genetic algorithm learning literature may be applied in a stochastic context.

Main findings We conduct a series of computational experiments with social learning in the setting studied by Bullard and Mitra (2002). Our main finding is that the Taylor Principle does not have to be met in order for agents to co-ordinate on a rational expectations minimum state variable (MSV) equilibrium of the model via the social learning dynamic. This stands in marked contrast to the findings in the recent learning literature.

This happens because social learning is actively trying to increase the fitness of its rules based on their forecasting error. When a rule that deviates from MSV appears in the population, this has a negative impact on its fitness value relatively to the other rules and is eventually, through selection of rules with higher fitness, driven out of the population.

We also find that agents cannot learn sunspot equilibrium in general representation but they can co-ordinate on sunspot solution in common factor (CF) representation.

Recent related literature Woodford (2003) contains the definitive statement of the nature of the New Keynesian model of monetary policy. Bullard (2006) surveys some of the literature on monetary policy and expectational stability, along with related issues. Genetic algorithm learning in economic contexts has been surveyed by Arifovic (2000).

Our article is related to the recent literature on heterogeneous learning. For example, Giannitsarou (2003) as well as Honkapohja and Mitra (2005, 2006) distin-

[2] See, for instance, Honkapohja and Mitra (2004).
[3] I.e. the set of problems which have been considered are deterministic, although the algorithm itself is necessarily stochastic.

guish the following forms of heterogeneity in learning: different initial perceptions, different learning rules and different degrees of inertia in updating the same learning rule. Giannitsarou (2003) finds that when agents use least squares learning, E-stability implies learnability in the case of different initial perceptions. However, for the other types of heterogeneity, the stability under homogeneous learning does not necessarily imply stability under heterogeneous learning. Our social learning approach encompasses a greater degree of heterogeneity than previous studies in this area, as a finite number of agents each have a different model within a given class of models.

Honkapohja and Mitra (2006) add structural heterogeneity to their analysis (agents respond differently to their forecasts) and study how transient and persistent heterogeneity in learning affects the learnability of the fundamental (MSV) solution. They find that transient heterogeneity in learning does not change the convergence conditions even in the presence of structural heterogeneity. However, in case of persistent heterogeneity in learning, E-stability conditions do not in general imply learnability in structurally homogeneous and heterogeneous economies. Honkapohja and Mitra (2005) study the performance of interest rate rules in the presence of heterogeneous forecasts by the private sector and the central bank in New Keynesian model. They find that E-stability conditions are necessary but not sufficient for learnability with heterogeneity in learning.

Negroni (2003) studies heterogeneity in adaptive expectations. He considers two sources: heterogeneity of expectations (different gains) and heterogeneity of fundamentals. He finds that in the presence of heterogeneity, the conditions for convergence of heterogeneous adaptive beliefs to the stationary REE are not the same as for homogeneous beliefs.

Our agents have the same form of the learning rule but different initial beliefs about the values of the coefficients in the perceived law of motion (PLM). They update their beliefs at the same rate, so the economy is structurally homogeneous. Our agents are able to learn from the other agents (social learning), whereas in all the models with heterogeneous learning mentioned above agents proceed to update their beliefs without knowing what and how well the rest of the agents are doing. The social aspect of the learning seems to be important for learning of the rational expectations equilibrium.

Branch and Evans (2006) show that heterogeneity can arise under certain conditions as an endogenous outcome when agents choose between misspecified models. In our study, agents have the correct specification of the REE model, although they start with different beliefs about the coefficients in the correct specification. Our question is whether agents are able to learn the fundamental (MSV) values of the coefficients.

In the version of New Keynesian model we study, sunspot equilibria exist in the indeterminate region where Taylor Principle is not satisfied. Sunspot equilibria can be represented in two forms – *general form with uncorrelated sunspots* and *CF form with sunspot that follows an autoregressive process*. Evans and McGough (2005*a,b*) find that a class of models and policy rules in which CF sunspots are learnable (even though general form sunspots are not learnable). However, they also find that for the model that we use in our article, neither general form nor CF representation sunspot equilibria are stable under recursive learning. We study whether sunspot equilibria are learnable under social learning and find that general form is not learnable but CF form is learnable.

In the next Section we discuss the New Keynesian model that we wish to study in this article. Much has been written about this model but here we only provide the reader with a minimal outline of the key equations, as the model itself is not the focus of this analysis. We then turn to a discussion of the social learning dynamic as we have implemented it in the New Keynesian model. Our main findings are the results of computational experiments, which we compare with standard results from the literature. Finally, we investigate if our agents can learn to co-ordinate on a sunspot equilibrium. We study both the general form and the CF specification of the sunspot equilibrium.

1. Environment

1.1. *Overview*

We study the simple version of the New Keynesian model employed by Bullard and Mitra (2002) and Woodford (2003). The economy is populated by a continuum of infinitely-lived household-firms that maximise utility and profits. Household-firms consume all goods but produce only one good on the continuum. Firms are monopolistically competitive and face a Calvo-style sticky price friction when determining their price. The model consists of three equations along with an exogenously specified stochastic process. The first equation is the linearised version of the first order condition for household utility maximisation. The second equation is the linearised version of the first order condition for firm maximisation of profits. The third equation is a Taylor-type interest rate feedback rule that describes the behaviour of the monetary authority.[4] The system is given by

$$z_t = z^e_{t+1} - \sigma^{-1}(r_t - \pi^e_{t+1}) + \sigma^{-1} r^n_t, \tag{1}$$

$$\pi_t = \kappa z_t + \beta \pi^e_{t+1}, \tag{2}$$

where z_t is the output gap, π_t is the deviation of the inflation rate from a pre-specified target, r_t is the deviation of the short-term nominal interest rate from the value that would hold in a steady state with the level of inflation at target, and output at the level consistent with fully flexible prices. A superscript e denotes a subjective expectation that can initially be different from a rational expectation. All variables are expressed in percentage point terms and the steady state is represented by the point $(z_t, \pi_t, r_t) = (0, 0, 0)$. The parameter $\beta \in (0,1)$ is the discount factor of the representative household, $\sigma > 0$ controls the intertemporal elasticity of substitution of the household and $\kappa > 0$ relates to the degree of price stickiness in the economy. A standard calibration suggested by Woodford (2003) and widely used in the literature sets $(\beta, \sigma, \kappa) = (0.99, 0.157, 0.024)$. The natural rate of interest, r^n_t, is a stochastic term which follows the process

$$r^n_t = \rho r^n_{t-1} + \epsilon_t, \tag{3}$$

[4] Optimal policy and learnability can also be studied, Evans and Honkapohja (2003).

where ϵ_t is *i.i.d.* noise with variance σ_ϵ^2, and $0 \leq \rho < 1$ is a serial correlation parameter. The interest rate feedback rule of the monetary authority is given by

$$r_t = \varphi_\pi \pi_t + \varphi_z z_t, \tag{4}$$

where φ_π and φ_z are policy parameters taken to be strictly positive. The policy maker is committed to this rule and does not deviate from it. Substituting (4) into (1), we obtain

$$z_t = z_{t+1}^e - \sigma^{-1}(\varphi_\pi \pi_t + \varphi_z z_t - \pi_{t+1}^e) + \sigma^{-1} r_t^n. \tag{5}$$

1.2. *Determinacy and Learnability*

Equations (2),(3) and (5) can be written as:

$$\boldsymbol{y}_t = \boldsymbol{\alpha} + \boldsymbol{B}\boldsymbol{y}_{t+1}^e + \boldsymbol{\chi} r_t^n, \tag{6}$$

where $\boldsymbol{\alpha} = \mathbf{0}$, $\boldsymbol{y}_t = [z_t, \pi_t]'$,

$$\boldsymbol{B} = \frac{1}{\sigma + \varphi_z + \kappa\varphi_\pi}\begin{bmatrix} \sigma & 1 - \beta\varphi_\pi \\ \kappa\sigma & \kappa + \beta(\sigma + \varphi_z) \end{bmatrix}, \tag{7}$$

and

$$\boldsymbol{\chi} = \frac{1}{\sigma + \varphi_z + \kappa\varphi_\pi}\begin{bmatrix} 1 \\ \kappa \end{bmatrix}. \tag{8}$$

To analyse the effects of homogeneous recursive learning in this environment, Bullard and Mitra (2002) proceeded as follows. Assume that all agents have the following PLM[5]

$$\boldsymbol{y}_t = \boldsymbol{a} + \boldsymbol{c} r_t^n, \tag{9}$$

which describes their belief concerning the equilibrium law of motion of the economy. With this PLM they form expectations as

$$E_t \boldsymbol{y}_{t+1} = \boldsymbol{a} + \boldsymbol{c}\rho r_t^n = \begin{bmatrix} z_{t+1}^e \\ \pi_{t+1}^e \end{bmatrix}. \tag{10}$$

The actual law of motion (ALM) is then found by substituting (10) into (6)

$$\boldsymbol{y}_t = \boldsymbol{B}\boldsymbol{a} + (\boldsymbol{B}\boldsymbol{c}\rho + \boldsymbol{\chi}) r_t^n. \tag{11}$$

The minimal state variable (MSV) solution is

$$\boldsymbol{y}_t = \bar{\boldsymbol{a}} + \bar{\boldsymbol{c}} r_t^n, \tag{12}$$

where $\bar{\boldsymbol{a}} = \mathbf{0}$ and $\bar{\boldsymbol{c}} = [\boldsymbol{I} - \rho\boldsymbol{B}]^{-1}\boldsymbol{\chi}$. At $(\bar{\boldsymbol{a}}, \bar{\boldsymbol{c}})$, the ALM coincides with the PLM and rational expectations equilibrium has been attained. If the ALM has dynamics which tend to this fixed point, we say that the equilibrium is learnable.

[5] The assignment of the PLM is not arbitrary but corresponds to the equilibrium law of motion of the economy.

Bullard and Mitra (2002) determine the necessary and sufficient condition for a rational expectations equilibrium to be determinate in the sense of Blanchard and Kahn (1980) as

$$\kappa(\varphi_\pi - 1) + (1 - \beta)\varphi_z > 0 \qquad (13).$$

Bullard and Mitra (2002) also show that this same condition is necessary and sufficient for the expectational stability of rational expectations equilibrium. Inequality (13) is a statement of the Taylor Principle. In particular, consider the simplified condition $\varphi_z = 0$, so that the central bank does not respond to deviations of output from potential when setting its nominal interest rate target. As $\kappa > 0$, the condition requires $\varphi_\pi > 1$, which is to say that the nominal interest rate must be adjusted more than one-for-one in response to deviations of inflation from target.

Bullard and Mitra (2002) concluded that condition (13) governs both uniqueness of rational expectations equilibrium as well as expectational stability of that equilibrium in this simple model.[6] Expectational stability is a notional time concept, but Evans and Honkapohja (2001) show that it governs the stability of the real time system formed when agents estimate the coefficients in (9) using recursive algorithms, such as least squares. We now turn to examine the robustness of this finding when homogeneous recursive learning is replaced with social learning.

2. Social Learning

2.1. *Overview*

We study the behaviour of evolutionary learning agents. Agents are initially heterogeneous with respect to their PLM (9), in the sense that each agent has a separate and possibly different set of coefficients. Thus each agent initially has a different forecasting model. The coefficients are updated using social evolutionary learning instead of least squares learning. Our objective is to see whether MSV solutions are learnable by evolutionary learning agents.

Social evolutionary learning is implemented using genetic algorithm. Genetic algorithm is a numerical optimisation technique, first introduced by Holland (1975) and described in Goldberg (1989), Michalewicz (1996), Back *et al.* (2000). There are several advantages of using genetic algorithm for optimisation. It starts with a set of random solutions and so does not rely on the starting point. It is a random search algorithm that prevents convergence to a suboptimal solution. It is applicable to discontinuous, non-differentiable, noisy, multi-modal and other unconventional surfaces (Schwefel, 2000). Genetic algorithms use stochastic transition rules to guide the search. While randomised, the algorithms are no simple random walk. They use random choice as a tool to guide a search towards regions of the search space with expected improved performance.

Genetic algorithms combine survival of the fittest with a structured yet randomised information exchange that resembles some of the innovative flair of human cognitive process.

[6] The relationship between determinacy and learnability is less clear in more complicated settings.

In economics literature, genetic algorithms have been primarily used to model economic agents adaptive behaviour.[7] The genetic algorithm is a 'model of innovation and creativity' (Dawid, 1999). Genetic algorithm describes a learning population where agents make decisions, try new ideas (mutation), exchange information (crossover) and learn from each other (selection). Genetic algorithms provide an environment that is very suitable for studying of the economies with heterogeneous agents. Unlike majority of other learning algorithms that have been used in macroeconomic environments, they do not impose high information and computational requirements on agents. Genetic algorithms have been used in different economic models. The examples include overlapping generations monetary economies (Arifovic, 1995; Dawid, 1996; Bullard and Duffy, 1998, 2001), foreign exchange (Arifovic, 1996; Lux and Schornstein, 2002), financial markets (LeBaron *et al.*, 1999; Lux and Marchesi, 2000; LeBaron, 2006), growth (Arifovic *et al.*, 1997), cobweb model (Arifovic, 1994; Dawid and Kopel, 1998; Franke, 1998; Vriend, 2000; Arifovic and Maschek, 2006), matching (Haruvy *et al.*, 2006), Walrasian general equilibrium (Gintis, 2007), public goods (Arifovic and Ledyard, 2010) and electricity markets (surveys in Sensfuß *et al.*, 2007; Weidlich and Veit, 2008).

2.2. *Initialisation*

We follow the artificial intelligence literature and take (1) and (2) as fixed and immutable in the analysis under evolutionary learning. In this interpretation, we are viewing the formal model with homogeneous expectations the only way it can be viewed, namely, as an approximation to a more realistic model with heterogeneous expectations.[8] The core model results of the previous Section can only be useful to the economics community if they are intended to be reasonably robust to the introduction of some heterogeneity among agents, especially with respect to agent expectations. We now introduce that heterogeneity.

There are N agents in the private sector. Each agent, $i = 1, \ldots, N$ has a PLM

$$z_t = a_{1,i,t} + c_{1,i,t} r_t^n, \tag{14}$$

$$\pi_t = a_{2,i,t} + c_{2,i,t} r_t^n. \tag{15}$$

The PLM has the same form as MSV equilibrium solution (12) but the parameter values in PLM can be different from REE values. Each agent i learns about vector of coefficients $(a_{1,i}, a_{2,i}, c_{1,i}, c_{2,i})$ that he revises in each period t. We stress that r_t^n is a stochastic term, and that finding equilibrium values of a and c will depend on evaluating how well each forecast rule works even though there is noise in the system. This is not a typical feature of evolutionary learning environments. It is true that the genetic operators we discuss below are inherently stochastic but the fitness calculation does not normally have to contend with exogenous stochastic terms.

The initial values for the coefficients are each randomly generated from a normal distribution with mean equal to the respective MSV value. The standard deviation for

[7] It has been also used as an optimisation technique. For example, Bullard and Duffy (2004) use simulated method of moments with a genetic algorithm to estimate growth model with structural breaks.

[8] For example, Branch and McGough (2009) determine restrictions under which (1) and (2) would arise as equilibrium conditions in a model with heterogeneous expectations.

coefficients c_1 and c_2 is equal the larger of the absolute values of the MSV values of these two coefficients. We used a smaller initial standard deviation for the coefficients a_1 and a_2. The MSV values for these coefficients are 0 and are smaller than MSV values for the coefficients c_1 and c_2. Therefore, we used initial standard deviation for coefficients a half as large as for the coefficients c. When setting the values of initial standard deviations we have pursued several objectives – starting with diverse population of rules, injecting diverse new rules through mutation and keeping the diversity of new rules commensurate with the MSV values.

We conducted sensitivity analysis about the value of standard deviation used for initialisation and mutation. When this value is twice as large or twice as small as baseline value, the performance is similar. But when this value is too high or too low (for example, above 10 times higher or lower, i.e. different by the order of magnitude), the performance of the algorithm deteriorates.[9] When the standard deviation of initialisation and mutation is too low, the diversity in the initial pool of rules is too low, and if the rules are away from MSV equilibrium, the diversifying force of mutation is too weak to be able to generate a transition towards the equilibrium. When the standard deviation is too high, the opposite is the issue – mutation becomes destructive by introducing new rules that deviate from equilibrium.

2.3. *Expectations and the ALM*

Agents form their expectations of the output gap and deviation of inflation from target using (3), (14), (15) as

$$z^e_{i,t+1} = a_{1,i,t} + c_{1,i,t}\rho r^n_t, \tag{16}$$

$$\pi^e_{i,t+1} = a_{2,i,t} + c_{2,i,t}\rho r^n_t. \tag{17}$$

The average expectations of the output gap and the deviation of inflation from target are computed as

$$z^e_{t+1} = \frac{1}{N}\sum_{i=1}^{N} z^e_{i,t+1}, \tag{18}$$

$$\pi^e_{t+1} = \frac{1}{N}\sum_{i=1}^{N} \pi^e_{i,t+1}. \tag{19}$$

The actual values of the output gap and deviation of inflation from target are obtained from:

$$\boldsymbol{y}_t = \boldsymbol{\alpha} + \boldsymbol{B}\begin{bmatrix} z^e_{t+1} \\ \pi^e_{t+1} \end{bmatrix} + \boldsymbol{\chi} r^n_t. \tag{20}$$

[9] These results are available upon request.

2.4. *Forecast Rule Performance*

Agents assess the performance, or *fitness*, of their forecasting model using mean squared forecast error as a criterion. Agents compute the mean squared forecast error for the output gap and the deviation of inflation over all periods following an initial history. We stress that it is important not to base the performance on only the most recent forecast error because the environment is stochastic.[10]

The fitness is computed as

$$F_{i,t} = -\frac{1}{t}\sum_{k=1}^{t}(z_k - z^f_{i,k})^2 - w\frac{1}{t}\sum_{k=1}^{t}(\pi_k - \pi^f_{i,k})^2, \tag{21}$$

where z^f_k is the forecast value of z for period k, π^f_k is the forecast value of π for period k and w is the relative weight on the mean squared error (MSE) for inflation. An agent is characterised by a set of coefficients $(a_{1,i,t},\ a_{2,i,t}\ ,\ c_{1,i,t},\ c_{2,i,t})$ at each date t. The terms z^f_k and π^f_k are the forecasts of the output gap and the deviation of inflation from target that agent i could have computed in period k, if he had used the current, date t, set of coefficients $(a_{1,i,t},\ a_{2,i,t},\ c_{1,i,t},\ c_{2,i,t})$. The forecasts z^f_k and π^f_k are computed by agent i as

$$z^f_{i,k} = a_{1,i,t} + c_{1,i,t}\rho r^n_{k-1}, \tag{22}$$

$$\pi^f_{i,k} = a_{2,i,t} + c_{2,i,t}\rho r^n_{k-1}. \tag{23}$$

The weight w is used to give equal importance to the prediction error for the output gap and the deviation of inflation from target as the values of the MSE for these two variables can differ in order of magnitude. Without reasonable weighting, the fitness measure puts insufficient emphasis on the first or the second term in (21), leading to drift in coefficients away from MSV values.

First, we considered simulations with weight $w = 1$, implying output forecast error volatility and inflation forecast error volatility have the same weight in the assessment of forecast rules.[11] From these simulations, we collected the data on fitness and its composition: the first and the second summation terms in (21). This data indicated that the MSE for z was several orders of magnitude larger than the MSE for π and therefore agents effectively did not care very much about the accuracy of their prediction for π when assessing their forecast rule. As a result, the coefficients diverged away from MSV values.[12]

The difference in magnitudes of MSE for output gap and MSE for inflation deviation can be explained by the difference in values of output gap and inflation deviations. From time series of z and π, we observed that output gap assumes larger values than inflation deviations. This comes from the values of coefficients in equation (20) for the computation of the actual output gap, z, and inflation deviation, π. At the standard calibration we use, the coefficients for the computation of z are several times larger than the coefficients for the computation of π. This makes values of z larger than values

[10] Branch and Evans (2006) assume that '... agents make their choices based on unconditional mean payoffs rather than on the most recent period's realized pay-off. This is more appropriate in a stochastic environment since otherwise agents would frequently be misled by single period anomalies'.

[11] The genetic operators used in these simulations are described below. Here, we simply wanted to discuss the fitness criterion.

[12] These results are available upon request.

of π, and so the squared prediction error for z larger than for π. In turn, this implies that in the fitness calculation, the first summation term in (21) is considerably larger than the second summation term (most frequently by a factor of 100). We used the weight w to adjust for this asymmetry. In particular, we set w such that the first and second summation terms in (21) are of the same order of magnitude. We use weight equal 100 for the simulations reported in this article.

The criterion (21) with $z_k^f = 0$ and $\pi_k^f = 0$ is a version of the objective function for the central bank that is often employed in models of optimal monetary policy. In studies of this type, w would represent the central bank's relative preference for inflation *versus* output volatility. This objective is also often rationalised as an approximation to the utility of the representative household in this economy, as suggested by Woodford (2003). In the optimal policy literature, w takes on a relatively large value. There the weight on inflation stabilisation is typically set to one, and the weight on output stabilisation is close to zero, so that the relative weight on inflation stabilisation is quite large. In the present article, the agents are concerned with the forecasting performance of their forecasting model, and so forecast performance matters and z_k^f as well as π_k^f are non-zero. However, the relatively large value of w that delivers the best performance of the learning model is similar.

2.5. *Genetic Operators*

A hallmark of the evolutionary learning literature is that agents update their current state using genetic operators. These operators are meant to simulate the exchange of information in a large, complex economy, and are based on the principles of population genetics. Agents can meet other agents, exchange information concerning their current forecast rule and possibly copy the partner's forecast rule, either in whole or in part. This process is implemented as described below.

We follow the literature in this area and use three genetic operators, namely crossover, mutation and tournament selection. Our genetic system is real-valued. The advantage of using real-valued system is its ability to deal with continuous search spaces and to be used for local tuning of the solutions (Herrera *et al.*, 1998).

Crossover is implemented first. It is applied to the pairs of agents (parents) and the outcome/offspring combines the characteristics of its parents. This can be interpreted as economic agents communicating with each other and adopting only part of the model of the other agent. Crossover is considered a powerful operator in the genetic algorithm literature. One is taking 'building blocks of good solutions' and combining them to create new possible solutions. This is thought to be a much faster way to find a good solution to a difficult problem than to merely rely on a mutation process. Especially for our real-valued, multi-dimensional problem, it can take a long time for mutation alone to find the best solution.

We implement simple crossover which is shown to perform generally well (Herrera *et al.* 1998). Two agents in the set of N agents are randomly matched without replacement. With probability of crossover *mcross*, their sets of coefficients are subjected to crossover: If a random draw from a uniform distribution is less than or equal to *mcross*, the agents exchange each type of coefficient with probability 0.5. Here is an example. These are two parent rules, $P1$ and $P2$, before crossover:

$$(a_1^{P1}, a_2^{P1}, c_1^{P1}, c_2^{P1}),$$
$$(a_1^{P2}, a_2^{P2}, c_1^{P2}, c_2^{P2}).$$

Suppose that coefficients a_2s are selected to be exchanged. The resulting rules after crossover become:

$$(a_1^{P1}, a_2^{P2}, c_1^{P1}, c_2^{P1}),$$
$$(a_1^{P2}, a_2^{P1}, c_1^{P2}, c_2^{P2}).$$

Depending on the outcome of random draws, more than one coefficient can be exchanged[13]. If coefficients a_2s and c_1s are chosen to be exchanged, the resulting rules after crossover become:

$$(a_1^{P1}, a_2^{P2}, c_1^{P2}, c_2^{P1}),$$
$$(a_1^{P2}, a_2^{P1}, c_1^{P1}, c_2^{P2}).$$

The above steps are implemented $N/2$ times.

Mutation is implemented following crossover. Mutation operator models introduction of new ideas, which can be good innovations or mistakes. This operator describes the observed behaviour where some good solutions are stepped on by chance, or occasionally mistakes are made. The mutation operator is important because it allows exploration of the entire search space and prevents premature convergence to sub-optimal solutions.

We implement mutation as follows. An agent changes each coefficient with probability of mutation *mprob* in the following way

$$new = old + random \times mutdeviation, \tag{24}$$

where *random* is a random number drawn from a standard normal distribution, *old* is the current value of the coefficient and *mutdeviation* is the standard deviation used for mutation. We set *mutdeviation* to be decreasing over time according to

$$mutdeviation = deviation \times (1 - decrease \times t/T), \tag{25}$$

where *deviation* is the standard deviation used to generate initial set of rules, t is current date, T is the total number of periods in the simulation and *decrease* is a coefficient. We set *decrease* equal to 0.95; it is intended to allow non-zero mutation standard deviation even in the last period of the simulation. Mutation can be very destructive late in a simulation when the N forecast rules may be very close to optimal REE forecast rules, because a random choice of a new coefficient will cause a new round of genetic variation. Equation (25) is meant to control this effect.

After mutation, agents compute the fitness of their coefficients according to (21).

The final genetic operator is tournament selection. Tournament selection describes behaviour where economic agents imitate more successful agents. In complex environments, it can be complicated to solve for the optimal solution. Imitation allows for better performing behaviours to be adopted by other agents. We implement tournament

[13] It is possible that no coefficients are exchanged, or, in the other extreme, all coefficients are exchanged (which means that two parent rules just changed their indices).

selection as follows. Agents are randomly selected in pairs with replacement N times. For each pair of agents, the fitness values of the forecast rules are compared. The agent with the higher fitness value is copied into the next generation of agents. This creates a new generation of N agents. After this update is finished, agents go to the next period of the simulation. Tournament selection will provide most of the selection pressure in this evolutionary learning environment as weaker forecasting rules are systematically discarded during this process.

3. Computational Experiments

3.1. *Overview*

We conduct a set of computational experiments to understand the behaviour of the economy under social learning. We begin our simulations by generating an initial history for the system at the rational expectations equilibrium, i.e. using the MSV values for the coefficients a and c. We then conduct simulations that last for 1,000 periods, and we set the length of the initial history to 100 periods. We use the parameter values from Woodford (2003), namely $\sigma = 0.157$, $\kappa = 0.024$, $\beta = 0.99$ and $\rho = 0.35$. The standard deviation of r^n is 3.72. We consider a range of values for the parameters in the Taylor-type monetary policy rule. In general, these are values of the coefficient on the output gap, $\varphi_z \in [0.2,\ 1.1]$. For the coefficient on inflation, we consider $\varphi_\pi \in [0.5,\ 2]$. At these parameter values, condition (13) is met for some policy parameter pairs but not for others, and is governed primarily by the value of φ_π. The value of w in the fitness criterion is 100.

We use the following parameter values in the genetic algorithm. The number of agents is 30. The probability of mutation is 0.1. High probability of mutation can introduce too much noise and it has been shown in the literature that low probability of mutation performs better. Therefore, we use low probability of mutation equal 0.1. The probability of crossover is 0.5. The related studies have shown that high probability of crossover performs better. We have performed the robustness checks for these parameters.

The role of crossover and mutation is to introduce new rules that can perform better. To illustrate the impact of crossover on the fitness of the forecasting rules, we measure the value of the maximum fitness before and after the application of the crossover and before and after the application of mutation. If both operators bring in new, good performing rules, this will be reflected in the increase in the maximum fitness and then these rules will be replicated through the application of the tournament selection. We find that both crossover and mutation improve the maximum fitness equally well and that this impact is insensitive to the choice of the probability of crossover. Higher probability of mutation increases chances of finding better rules and so we observe larger improvement in maximum fitness with increase in the mutation rate. However, higher mutation probability can be destructive too. We find that low mutation probability (0.1) generates the best improvements in average fitness.[14] For similar results about desirability of low mutation probability and insensitivity to probability of crossover, Lux and Schornstein (2002).

[14] The results of the robustness tests for both crossover and mutation are available upon request.

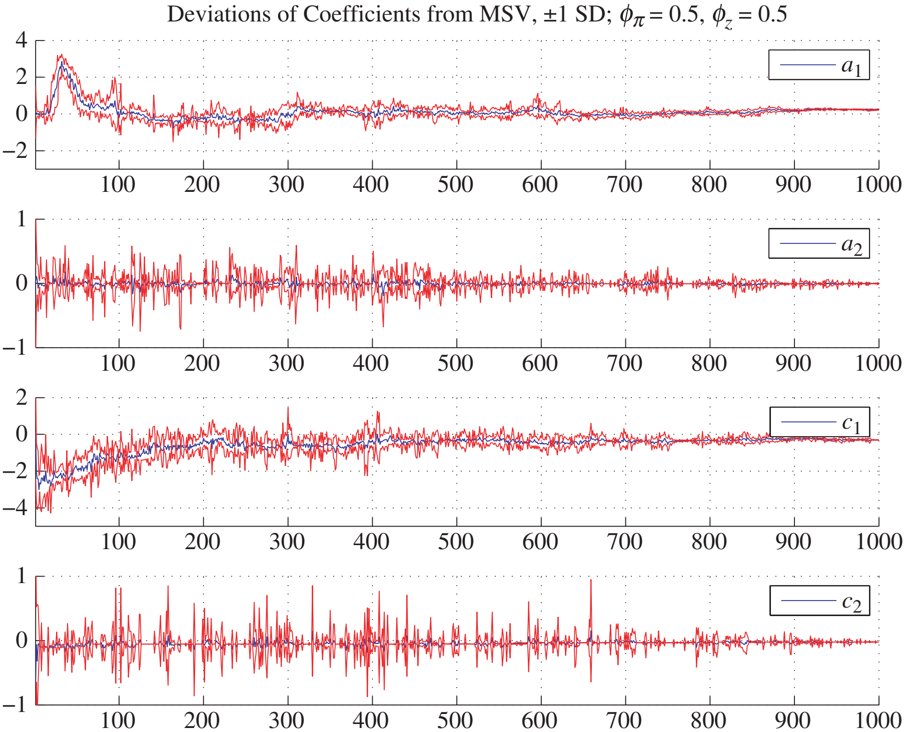

Fig. 1. *Simulation for Determinate and E-stable Region:* $\phi_\pi = 2$, $\phi_z = 0.2$

3.2. *Main Findings*

We found that agents are able to learn MSV values of coefficients for most of the policy parameter pairs (φ_z, φ_π), both in determinate and E-stable region as well as in the indeterminate and E-unstable region. A series of four figures shows our main results.

A typical simulation result for the policy rule characterised by $\varphi_\pi = 2.0$ and $\varphi_z = 0.2$ is given in Figure 1 and, for the policy rule $\varphi_\pi = 1.5$ and $\varphi_z = 0.5$, in Figure 2. These policy rules are associated with a determinate rational expectations equilibrium and E-stability. The Figures show the time series of the deviation of each of the four coefficients from their MSV values averaged across all agents. The Figure also shows ±1 standard deviation for each coefficient's deviation from MSV values, showing the extent of the dispersion in coefficients in use at date t in the population of agents. Figures 1 and 2, along with other simulations using policy rules consistent with determinacy and learnability, suggest that long-run predictions from analyses using recursive learning and analyses using evolutionary learning are similar. In particular, both approaches predict convergence to the rational expectations equilibrium.[15]

This result breaks down when we consider other policy rules, however.

[15] It should be noted that 'social learning' convergence result refers to the convergence in simulations. We tested our results conducting our simulations for 10,000 periods. We have not observed any change or difference from what is reported in the article. We do not report these results in detail but they are available upon a request.

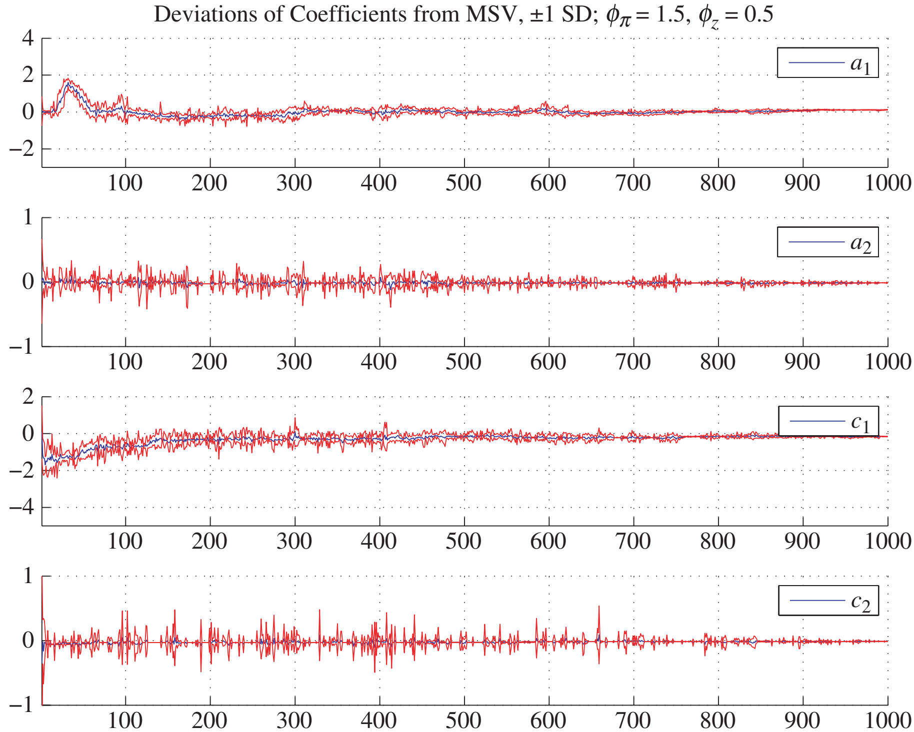

Fig. 2. *Simulation for Determinate and E-stable Region:* $\phi_\pi = 1.5,\ \phi_z = 0.5$

Figures 3 and 4 show typical simulation results for $\varphi_\pi = 0.5$ and $\varphi_z = 0.5$ or $\varphi_z = 0.3$ respectively. These policy rules are associated with indeterminacy and expectational instability. These Figures again show the time series of the deviation of each of the four coefficients from their MSV values averaged across all agents, and ±1 standard deviation. Here, the evolutionary learning dynamic converges in simulation to the MSV solution once again, even though the analysis based on least squares learning would predict instability in the learning dynamics. These findings suggest that, provided one is willing to take an evolutionary learning perspective, the less aggressive policy rules are not as disturbing as they may have appeared to be.

To provide more details concerning these results, we performed 1,000 simulations for different policy rules (φ_π, φ_z) and collected data for the deviations of coefficients from their MSV values for each simulation. During each simulation, for each coefficient, we computed the average value of deviation from the MSV value for each period. Then we computed average value of the average deviations during the last 100 periods of simulation.[16] We also compute average of absolute values of deviations from MSV values during last 100 periods. In addition, we collected data on percentage deviations from MSV values for coefficients c_1 and c_2 and computed average of (absolute) percentage deviations during last 100 periods of the simulations. (We cannot compute

[16] The results are not qualitatively different for data computed for last 100-period, last 10-period and last 1-period; therefore, we only report results for last 100-period data.

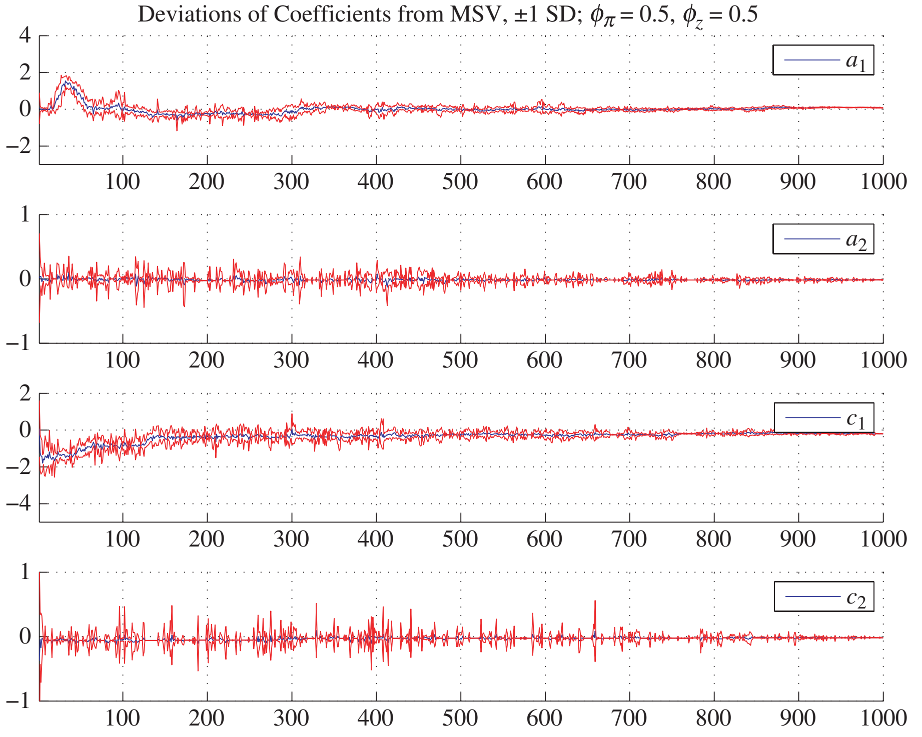

Fig. 3. *Simulation for Indeterminate and E-unstable Region:* $\phi_\pi = 0.5$, $\phi_z = 0.5$

percentage deviations for coefficients a_1 and a_2 as their MSV values are zero). For each policy rule (φ_π, φ_z), we perform 1,000 simulations, collect the above described statistics for each simulation and report means and standard deviations for each statistic over 1,000 simulations.

Table 1 reports means and standard deviations for average deviations from MSV values for a variety of fixed policy rules. The policy rules presented in this Table include some that induce a determinate and E-stable rational expectations equilibrium, as well as others that induce indeterminacy and expectational instability. The policy rules that induce determinacy and learnability according to condition (13) will have larger values of φ_π and φ_z, which tend to be located towards the northeast part of the Table. Relatively small values for φ_π and φ_z are associated with indeterminacy and expectational instability, and tend to be located in the southwest portion of the Table.

We can make the following observations from Table 1. Perhaps most importantly, for the policy rules considered, regardless of whether they are consistent with determinacy and learnability or not, the population coefficients are quite close to their MSV values. The genetic algorithm we have implemented allows mutation up to date T in the simulation and so does not attempt to eliminate variation entirely, yet the Table indicates that the population is quite close to the one that would use MSV values exclusively (all values in the Table are very close to zero). To the extent there are differences from MSV values, the deviations for the constant coefficients a_1 and a_2 can

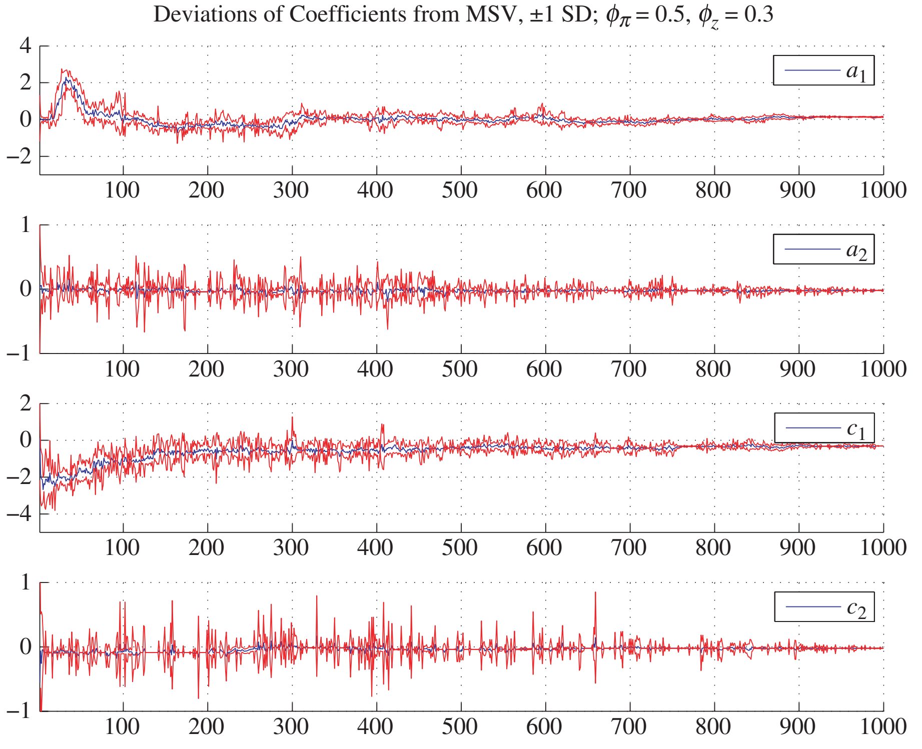

Fig. 4. *Simulation for Indeterminate and E-unstable Region:* $\phi_\pi = 0.5$, $\phi_z = 0.3$

be somewhat higher than those for the slope coefficients c_1 and c_2. Standard deviations indicate that there is some variety in the population even during the last 100 periods of the simulation but the extent of the variety is not very large.

Table 2 presents means and standard deviations for absolute values of the deviations from the MSV values. This Table also presents the percentage absolute deviation for the slope coefficients c_1 and c_2. These percentages for the absolute deviations range from about 3.0 to 11.0, and do not seem to vary systematically with the policy rule.

The previous Tables illustrate convergence in simulation of each individual coefficient. We would also like to present a measure of convergence for a complete set of coefficients – how close all coefficients are to MSV values at the same time. Table 3 reports the number of simulations out of 1,000 that satisfy specific convergence criteria based on averages of absolute deviations over last 100 periods of simulation. As different coefficients deviate from MSV values by different values, we present results of application of two criteria for convergence. Criterion 1 requires that absolute deviations from MSV values for all coefficients are less than or equal to 0.2. Criterion 2 requires that the absolute deviation from the MSV value for a_1 is less than or equal to 0.5 and that the absolute deviations from the MSV values for a_2, c_1 and c_2 are less than or equal to 0.3.[17]

[17] The number of simulations satisfying criterion 2 is very close to the number of simulations satisfying a criterion which requires that the absolute value of the deviation of coefficient c_2 from its MSV value is less than 0.03, and the rest of the coefficients satisfy criterion 2.

Table 1

Averages and Standard Deviations (SD) of Coefficient Deviations

Parameter		0.5		1.0		1.5		2.0	
φ_z	φ_π	Mean	SD	Mean	SD	Mean	SD	Mean	SD
1.1	a_1	−0.001	0.078	0.000	0.063	0.001	0.066	0.001	0.078
	a_2	−0.002	0.081	−0.001	0.076	−0.002	0.071	−0.002	0.066
	c_1	−0.002	0.044	−0.001	0.043	0.000	0.042	0.001	0.042
	c_2	−0.002	0.003	−0.002	0.003	−0.002	0.003	−0.002	0.003
1.0	a_1	−0.001	0.089	0.000	0.070	0.001	0.073	0.002	0.089
	a_2	−0.002	0.089	−0.001	0.082	−0.002	0.077	−0.002	0.071
	c_1	−0.002	0.048	−0.001	0.047	0.000	0.046	0.001	0.045
	c_2	−0.002	0.003	−0.002	0.003	−0.002	0.003	−0.002	0.003
0.9	a_1	−0.001	0.102	0.000	0.078	0.001	0.082	0.002	0.103
	a_2	−0.002	0.098	−0.002	0.090	−0.002	0.084	−0.002	0.078
	c_1	−0.002	0.053	−0.001	0.052	0.000	0.051	0.001	0.050
	c_2	−0.002	0.004	−0.002	0.004	−0.002	0.004	−0.002	0.003
0.8	a_1	−0.001	0.120	0.000	0.088	0.001	0.094	0.003	0.121
	a_2	−0.002	0.110	−0.002	0.100	−0.003	0.092	−0.002	0.084
	c_1	−0.003	0.060	−0.001	0.058	0.000	0.056	0.001	0.055
	c_2	−0.003	0.004	−0.003	0.004	−0.003	0.004	−0.002	0.004
0.7	a_1	−0.001	0.146	0.000	0.101	0.002	0.111	0.003	0.145
	a_2	−0.002	0.125	−0.002	0.113	−0.003	0.102	−0.002	0.092
	c_1	−0.004	0.068	−0.002	0.066	0.000	0.064	0.002	0.062
	c_2	−0.003	0.005	−0.003	0.005	−0.003	0.004	−0.003	0.004
0.6	a_1	−0.002	0.183	0.000	0.120	0.003	0.134	0.003	0.182
	a_2	−0.002	0.145	−0.003	0.129	−0.003	0.115	−0.002	0.103
	c_1	−0.005	0.078	−0.002	0.075	0.000	0.073	0.002	0.070
	c_2	−0.004	0.005	−0.003	0.005	−0.003	0.005	−0.003	0.005
0.5	a_1	−0.002	0.240	0.000	0.146	0.003	0.168	0.004	0.234
	a_2	−0.003	0.170	−0.003	0.150	−0.003	0.131	−0.002	0.115
	c_1	−0.007	0.092	−0.003	0.088	0.000	0.085	0.003	0.082
	c_2	−0.004	0.006	−0.004	0.006	−0.004	0.006	−0.003	0.005
0.4	a_1	−0.003	0.343	0.001	0.186	0.005	0.222	0.005	0.318
	a_2	−0.003	0.212	−0.004	0.178	−0.003	0.152	−0.002	0.131
	c_1	−0.010	0.112	−0.004	0.107	0.000	0.102	0.004	0.097
	c_2	−0.005	0.008	−0.005	0.007	−0.004	0.007	−0.004	0.007
0.3	a_1	−0.002	0.548	0.003	0.256	0.008	0.320	0.010	0.463
	a_2	−0.003	0.280	−0.004	0.221	−0.004	0.182	−0.002	0.151
	c_1	−0.016	0.144	−0.007	0.135	0.001	0.127	0.007	0.121
	c_2	−0.007	0.010	−0.006	0.009	−0.006	0.008	−0.005	0.008
0.2	a_1	−0.004	1.051	0.007	0.405	0.017	0.523	0.016	0.735
	a_2	−0.005	0.399	−0.004	0.291	−0.005	0.226	−0.002	0.173
	c_1	−0.030	0.200	−0.013	0.183	0.001	0.170	0.011	0.158
	c_2	−0.010	0.014	−0.008	0.012	−0.007	0.011	−0.006	0.010

Table 3 perhaps indicates a result more in conformity with previous findings in the learning literature: The number of simulations out of 1,000 satisfying either convergence criterion clearly tends to decline as one moves towards the southwest in Table 3, i.e. as one moves towards the region of the parameter space which is associated with indeterminacy and expectational instability. This is perhaps clearest when comparing

Table 2
Averages and Standard Deviations (SD) for Absolute Values of Coefficient Deviations

Parameter		0.5		1.0		1.5		2.0	
φ_z	φ_π	Mean	SD	Mean	SD	Mean	SD	Mean	SD
1.1	a_1	0.061	0.049	0.050	0.039	0.052	0.040	0.059	0.051
	a_2	0.046	0.067	0.043	0.062	0.041	0.059	0.038	0.054
	c_1	0.035	0.027	0.035	0.026	0.034	0.025	0.033	0.025
	c_2	0.003	0.002	0.003	0.002	0.003	0.002	0.003	0.002
	perc c_1	4.265	3.226	4.236	3.204	4.205	3.172	4.178	3.165
	perc c_2	10.129	6.779	10.015	6.833	9.992	6.717	9.819	6.478
1.0	a_1	0.069	0.056	0.055	0.043	0.058	0.045	0.066	0.059
	a_2	0.050	0.073	0.047	0.067	0.044	0.063	0.041	0.058
	c_1	0.039	0.029	0.038	0.028	0.037	0.028	0.036	0.027
	c_2	0.003	0.002	0.003	0.002	0.003	0.002	0.003	0.002
	perc c_1	4.280	3.238	4.253	3.206	4.213	3.196	4.183	3.177
	perc c_2	10.145	6.843	9.937	6.763	9.967	6.699	9.873	6.533
0.9	a_1	0.079	0.065	0.062	0.047	0.065	0.051	0.076	0.070
	a_2	0.056	0.081	0.052	0.074	0.048	0.069	0.045	0.063
	c_1	0.043	0.032	0.042	0.031	0.040	0.031	0.040	0.030
	c_2	0.004	0.002	0.004	0.002	0.004	0.002	0.003	0.002
	perc c_1	4.294	3.254	4.272	3.227	4.224	3.208	4.210	3.190
	perc c_2	10.231	6.818	10.059	6.785	10.048	6.703	9.814	6.532
0.8	a_1	0.093	0.077	0.070	0.054	0.074	0.059	0.088	0.083
	a_2	0.063	0.090	0.058	0.082	0.053	0.076	0.048	0.068
	c_1	0.048	0.036	0.046	0.035	0.045	0.034	0.044	0.033
	c_2	0.004	0.003	0.004	0.003	0.004	0.003	0.004	0.002
	perc c_1	4.339	3.275	4.285	3.235	4.247	3.222	4.226	3.198
	perc c_2	10.284	6.834	10.048	6.883	9.981	6.842	9.700	6.530
0.7	a_1	0.111	0.094	0.080	0.062	0.086	0.070	0.104	0.101
	a_2	0.072	0.103	0.065	0.092	0.059	0.084	0.053	0.075
	c_1	0.054	0.041	0.052	0.040	0.051	0.038	0.049	0.037
	c_2	0.005	0.003	0.005	0.003	0.004	0.003	0.004	0.003
	perc c_1	4.371	3.307	4.320	3.269	4.278	3.235	4.250	3.217
	perc c_2	10.408	7.025	10.180	6.934	10.085	6.770	9.690	6.552
0.6	a_1	0.138	0.120	0.095	0.073	0.103	0.085	0.128	0.130
	a_2	0.083	0.119	0.074	0.105	0.067	0.094	0.059	0.084
	c_1	0.062	0.047	0.060	0.046	0.058	0.044	0.056	0.042
	c_2	0.005	0.004	0.005	0.003	0.005	0.003	0.005	0.003
	perc c_1	4.411	3.336	4.362	3.308	4.307	3.254	4.284	3.231
	perc c_2	10.443	7.044	10.286	6.908	10.093	6.750	9.645	6.479
0.5	a_1	0.180	0.159	0.115	0.089	0.128	0.109	0.160	0.171
	a_2	0.098	0.139	0.087	0.122	0.076	0.107	0.067	0.094
	c_1	0.074	0.056	0.071	0.053	0.068	0.051	0.065	0.049
	c_2	0.006	0.004	0.006	0.004	0.006	0.004	0.005	0.004
	perc c_1	4.484	3.375	4.418	3.342	4.356	3.303	4.316	3.278
	perc c_2	10.522	7.044	10.312	6.918	10.172	6.816	9.701	6.569
0.4	a_1	0.254	0.230	0.147	0.113	0.166	0.147	0.212	0.237
	a_2	0.123	0.173	0.104	0.145	0.090	0.123	0.077	0.106
	c_1	0.090	0.068	0.085	0.065	0.081	0.062	0.077	0.059
	c_2	0.008	0.005	0.007	0.005	0.007	0.005	0.006	0.004
	perc c_1	4.567	3.453	4.487	3.398	4.409	3.350	4.353	3.309
	perc c_2	10.941	7.189	10.346	6.994	10.188	6.845	9.816	6.710

Table 2
(Continued)

Parameter		0.5		1.0		1.5		2.0	
φ_z	φ_π	Mean	SD	Mean	SD	Mean	SD	Mean	SD
0.3	a_1	0.395	0.379	0.204	0.154	0.233	0.219	0.300	0.352
	a_2	0.162	0.228	0.130	0.179	0.107	0.147	0.090	0.121
	c_1	0.116	0.087	0.108	0.082	0.101	0.077	0.096	0.073
	c_2	0.010	0.007	0.009	0.006	0.008	0.006	0.008	0.005
	perc c_1	4.716	3.552	4.584	3.473	4.504	3.410	4.463	3.360
	perc c_2	11.160	7.344	10.716	7.166	10.165	6.796	9.596	6.470
0.2	a_1	0.739	0.747	0.324	0.244	0.369	0.371	0.462	0.571
	a_2	0.234	0.323	0.174	0.233	0.137	0.180	0.107	0.136
	c_1	0.162	0.122	0.146	0.111	0.135	0.103	0.126	0.095
	c_2	0.014	0.009	0.012	0.008	0.011	0.007	0.010	0.007
	perc c_1	4.972	3.752	4.774	3.609	4.649	3.544	4.570	3.462
	perc c_2	11.647	7.625	10.918	7.340	10.204	6.906	9.753	6.623

Table 3
Convergence Criteria

Parameter	φ_π	0.5	1.0	1.5	2.0
φ_z	Criterion				
1.1	1	950	960	964	963
	2	990	992	994	995
1.0	1	933	947	955	957
	2	982	990	989	991
0.9	1	901	931	945	935
	2	973	981	989	991
0.8	1	866	911	921	902
	2	967	972	979	987
0.7	1	810	863	880	860
	2	952	962	971	974
0.6	1	725	801	822	807
	2	932	945	961	966
0.5	1	604	694	739	717
	2	901	925	944	938
0.4	1	453	554	607	619
	2	830	888	919	889
0.3	1	280	379	467	490
	2	668	800	847	815
0.2	1	145	208	275	306
	2	403	578	677	690

the most northeasterly cell in the Table with the cell in the southwest corner. The former is associated with determinacy and expectational stability, whereas the latter is not. In the northeast corner we observe values of 963 and 995, respectively, for the two convergence criteria, whereas in the southwest corner we observe values of 145 and 403. This would seem to be a clear indication that it is somehow 'more difficult' for the social learning system to co-ordinate upon the MSV solution, when expectational stability and determinacy conditions fail. However, we do not wish to press this point too hard. The cell associated with $\varphi_\pi = 1.0$ and $\varphi_z = 0.2$ has values of 208 and 578 for the two convergence criteria, respectively, not very different from the results for the cell in the southwest corner. Yet these parameter values satisfy condition (13); rational expectations equilibrium here is unique and expectationally stable. One other point is that Tables 1 and 2 indicated that whatever failure to co-ordinate may exist, actual values are not very different from MSV values, and would probably not be meaningful in economic terms.

In some simulations, we can observe deviations of average values of coefficients a_1 and a_2 from their MSV counterparts, even though agents are always able to learn MSV values of c_1 and c_2 quite closely. Again considering Table 2, to the extent that agents are inaccurate in learning MSV values, it is because of the coefficients a_1 and a_2, as the deviation of these coefficients from MSV is the largest among all coefficients. In the least squares learning model of Bullard and Mitra (2002), as pointed by Woodford (2003), '... it is in fact the possible instability of the dynamics of estimates of the constant terms Γ_0 in the forecasting model that is the relevant threat; and whether this occurs or not is determined by whether or not the Taylor principle is adhered to ...' (Woodford, 2003, pp. 271–2). In our notation, Γ_0 corresponds to the coefficients a_1 and a_2. Similarly, Honkapohja and Mitra (2004) point out that 'In Bullard and Mitra (2002), the constant term was the key to E-stability of the MSV solution ...' (Honkapohja and Mitra, 2004, p. 1757). However, our simulations show that the system under evolutionary learning behaves somewhat differently. While the values of a_1 and a_2 may not be as close to their MSV values as the values of c_1 and c_2, this effect occurs whether or not the Taylor Principle holds.

We have also explored the impact of the values of coefficients in the Taylor rule on the volatility of output gap and deviations of inflation from the target, as the stability of these macroeconomic variables may be important to the policy maker. We have computed standard deviations of output gap, z, and deviations of inflation from target, π, for each parameter set (based on 100 simulations). The statistics are presented in Table 4. We observe that volatility increases as coefficients ϕ_z and ϕ_π decrease and the system

Table 4
Standard Deviations (SD) of z and π

Parameter		0.5		1.0		1.5		2.0	
φ_z	φ_π	Average	SD	Average	SD	Average	SD	Average	SD
1.1	z	3.277	0.096	3.230	0.093	3.184	0.093	3.139	0.094
	π	0.121	0.027	0.119	0.025	0.118	0.024	0.116	0.023
0.8	z	4.357	0.132	4.275	0.128	4.195	0.127	4.116	0.129
	π	0.161	0.036	0.158	0.033	0.155	0.031	0.154	0.033
0.5	z	6.503	0.215	6.319	0.202	6.146	0.198	5.978	0.204
	π	0.243	0.055	0.234	0.049	0.227	0.046	0.224	0.048
0.2	z	12.784	0.568	12.103	0.472	11.485	0.454	10.924	0.473
	π	0.476	0.118	0.447	0.099	0.425	0.093	0.407	0.085

moves towards indeterminate region. However, it is not always the case that volatility is higher in indeterminate region than volatility in the determinate region. For example, for parameter set $\phi_\pi = 2$, $\phi_z = 0.2$ in the determinate region (the simulation for this parameter set is presented on Figure 1) the volatility is higher than that for parameter set $\phi_\pi = 0.5$, $\phi_z = 0.5$ in the indeterminate region (the simulation for this parameter set is presented on Figure 3). Thus, some rules that satisfy the Taylor Principle can lead to a higher volatility that rules that do not satisfy it.

3.3. *Different Performance Evaluation*

As we stressed earlier, the weighting of the two dimensions in the fitness criterion is essential to convergence of the social learning systems we study. Without reasonable weighting, the fitness measure puts insufficient emphasis on one dimension or the other, leading to drift in coefficients away from MSV values. The modification considered in this section has each agent compute the mean squared error for forecasting deviation of inflation from target and the output gap separately, and simply consider them separately without combining them into one fitness measure. In particular, agent i computes mean squared errors for the output gap and inflation as

$$F^z_{i,t} = -\frac{1}{t}\sum_{k=1}^{t}(z_k - z^f_{i,k})^2, \tag{26}$$

$$F^\pi_{i,t} = -\frac{1}{t}\sum_{k=1}^{t}(\pi_k - \pi^f_{i,k})^2, \tag{27}$$

where $z^f_{i,k}$, $\pi^f_{i,k}$ are computed as in (22) and (23).

The change in performance criterion also has effects on the tournament selection operator. We modified the operator as follows. Again, N pairs of agents are randomly selected from the current generation with replacement and fitness is compared for each pair. A new member of the next generation adopts the coefficients for forecasting output gap from the agent with higher $F^z_{i,t}$ (lower mean squared error for forecasting the output gap) and the coefficients for forecasting the deviation of inflation from the target from the agent with higher $F^\pi_{i,t}$ (lower mean squared error for forecasting inflation). In this way the next generation of agents is created, and more fit forecasting rules are systematically selected while weaker rules are systematically discarded.[18]

The results of these simulations are reported in Table 5. This Table reports the same data as Table 2 for the baseline simulations. The results are qualitatively the same as for the baseline simulations. Table 6 reports the number of simulations that satisfy convergence criteria. We find similar effects when moving from northeast to southwest in this Table as we did in Table 3.

[18] One of the principles of the GA literature is to accomplish this task without losing genetic diversity too rapidly.

Table 5

Averages and Standard Deviations (SD) for Absolute Values of Coefficient Deviations, Separate Fitness

Parameter		0.5		1.0		1.5		2.0	
φ_z	φ_π	Mean	SD	Mean	SD	Mean	SD	Mean	SD
1.1	a_1	0.064	0.053	0.050	0.039	0.054	0.044	0.066	0.062
	a_2	0.061	0.083	0.057	0.078	0.053	0.072	0.050	0.066
	c_1	0.034	0.027	0.033	0.026	0.033	0.026	0.032	0.025
	c_2	0.003	0.002	0.003	0.002	0.003	0.002	0.003	0.002
	perc c_1	4.117	3.230	4.091	3.199	4.059	3.187	4.036	3.168
	perc c_2	9.102	6.030	8.946	6.035	8.812	5.899	8.687	5.867
0.8	a_1	0.099	0.086	0.069	0.055	0.079	0.068	0.100	0.102
	a_2	0.084	0.114	0.077	0.103	0.070	0.093	0.063	0.084
	c_1	0.046	0.036	0.045	0.035	0.044	0.034	0.042	0.033
	c_2	0.004	0.002	0.004	0.002	0.003	0.002	0.003	0.002
	perc c_1	4.189	3.280	4.149	3.250	4.112	3.224	4.081	3.204
	perc c_2	9.208	6.131	9.037	6.042	8.893	5.929	8.678	5.868
0.5	a_1	0.202	0.190	0.115	0.091	0.141	0.130	0.188	0.211
	a_2	0.133	0.179	0.116	0.154	0.100	0.130	0.087	0.113
	c_1	0.071	0.056	0.068	0.054	0.065	0.051	0.063	0.049
	c_2	0.006	0.004	0.005	0.004	0.005	0.003	0.005	0.003
	perc c_1	4.330	3.400	4.262	3.349	4.208	3.303	4.151	3.263
	perc c_2	9.469	6.365	9.200	6.211	8.946	5.981	8.759	5.936
0.2	a_1	0.886	0.938	0.325	0.258	0.417	0.460	0.567	0.715
	a_2	0.314	0.410	0.236	0.301	0.178	0.218	0.138	0.164
	c_1	0.156	0.122	0.141	0.111	0.130	0.102	0.121	0.095
	c_2	0.012	0.008	0.011	0.007	0.010	0.007	0.009	0.006
	perc c_1	4.794	3.756	4.608	3.627	4.477	3.520	4.380	3.442
	perc c_2	10.346	6.848	9.801	6.478	9.243	6.309	8.841	6.074

Table 6

Convergence Criteria, Separate Fitness

Parameter	φ_π	0.5	1.0	1.5	2.0
φ_z	Criterion				
1.1	1	921	941	952	949
	2	975	980	984	987
0.8	1	828	874	893	869
	2	946	959	968	974
0.5	1	553	669	710	682
	2	861	898	924	920
0.2	1	127	193	267	275
	2	371	549	620	619

4. Comparison to Other Learning Algorithms

In this Section, we discuss the difference between the social learning algorithm and the algorithm commonly used in macroeconomic learning literature, constant gain least squares algorithm. From the literature on expectational stability, we know that the least squares algorithm does not converge if the Taylor Principle is not satisfied. To illustrate this, we compare the dynamics generated in simulations with constant gain least squares learning to those generated in simulations with social learning. We simulate the behaviour of the least squares updating for a set of parameter values from the region that is not E-stable and compare it with the behaviour of social learning.

We implement constant gain least squares learning of our model in the following way. The agents know the correct specification of the solution, but do not know the equilibrium values of the coefficients. The PLM is similar to that in social learning (14) and (15) and is given as:

$$z_t = a_{1,t} + c_{1,t} r_t^n, \tag{28}$$

$$\pi_t = a_{2,t} + c_{2,t} r_t^n. \tag{29}$$

In period t, the forecasts are made according to (similarly to 16 and 17).

$$\begin{aligned} z_{t+1}^e &= a_{1,t} + c_{1,t} r_t^n \rho, \\ \pi_{t+1}^e &= a_{2,t} + c_{2,t} r_t^n \rho. \end{aligned}$$

The forecasts are used to compute actual values according to (20).

In general, using the least squares scheme, the coefficients are updated according to:

$$\boldsymbol{\beta}_t = \boldsymbol{\beta}_{t-1} + \gamma \boldsymbol{R}_t^{-1} \boldsymbol{x}_t (\boldsymbol{y}_t - \boldsymbol{x}_t' \boldsymbol{\beta}_{t-1}), \tag{30}$$

$$\boldsymbol{R}_t = \boldsymbol{R}_{t-1} + \gamma (\boldsymbol{x}_t \boldsymbol{x}_t' - \boldsymbol{R}_{t-1}), \tag{31}$$

where $\boldsymbol{\beta}_t$ and $\boldsymbol{R}_t$ denote the coefficient vector and the moment matrix for x_t using data $i = 1,\ldots, t$.

Parameter γ is called gain, and we use constant value of 0.2.[19] We initialise all coefficients and moment matrix using the data of initial history that are simulated as MSV equilibrium (Evans and Honkapohja, 2001, pp. 32–3).

The deviations of coefficients from their MSV equilibrium values are depicted on Figure 5 for indeterminate and E-unstable region. We can see that all coefficients a_1, a_2 and c_2 diverge from their MSV values. Figure 6 shows fitness of the ‘rule’ updated with constant gain least squares learning, the value of fitness is calculated the same way as for the social learning algorithm in (21). We can see that fitness declines when coefficients diverge from their MSV values.

We also compute fitness in the simulations with social learning. Figure 7 illustrates typical behaviour of average fitness of agents in the simulation with social learning presented earlier on Figure 3. The average fitness drops at the beginning, but then increases by the end of the simulation. The fitness is very volatile at the beginning of the simulation: the sudden drops in fitness are because of the changes in the individual rules caused by the mutation (as seen by the increased standard deviations of coefficients at the same time on Figure 8).

Figures 6 and 7 nicely illustrate the difference in the way the updating works in case of constant gain least squares learning on one hand and social learning on the other hand. Social learning is driven by evaluation of fitness. As soon as the coefficients in the

[19] We have obtained qualitatively similar results for decreasing gain $\gamma = 1/t$.

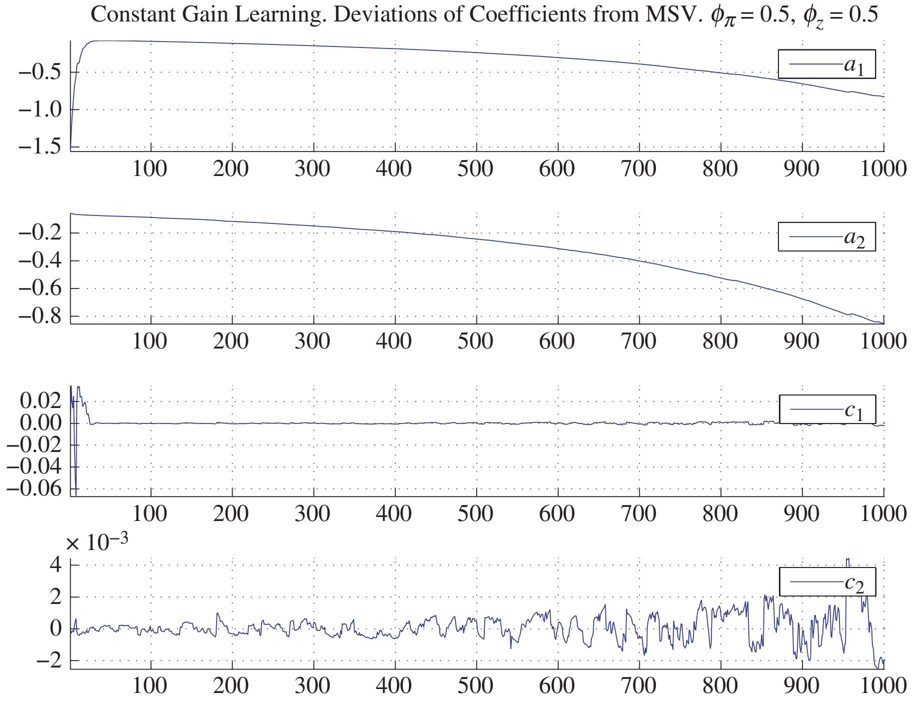

Fig. 5. *Constant Gain Learning in Indeterminate and E-unstable Region:* $\phi_\pi = 0.5$, $\phi_z = 0.5$.

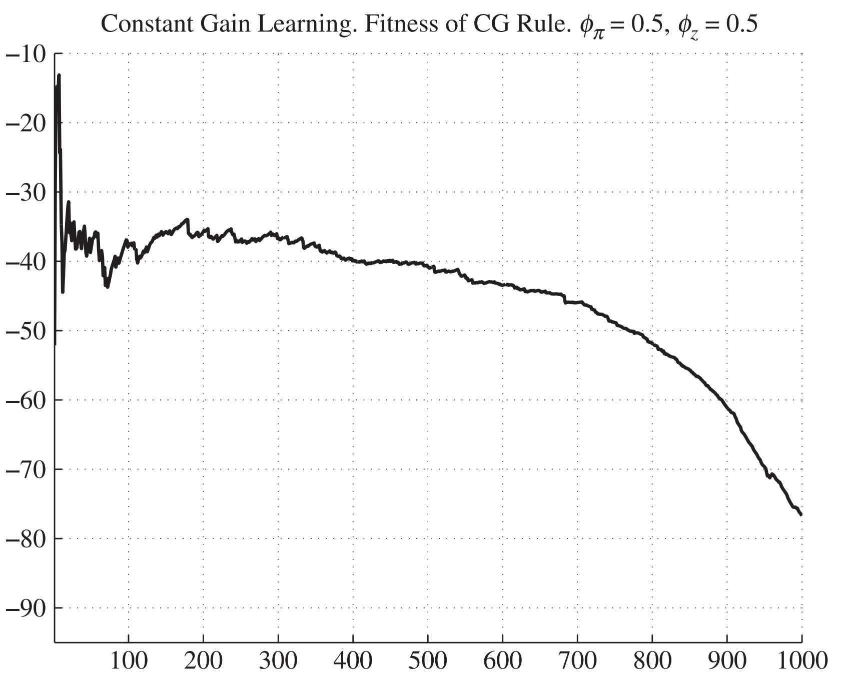

Fig. 6. *Fitness, Constant Gain Learning, Indeterminate and E-unstable Region:* $\phi_\pi = 0.5$, $\phi_z = 0.5$

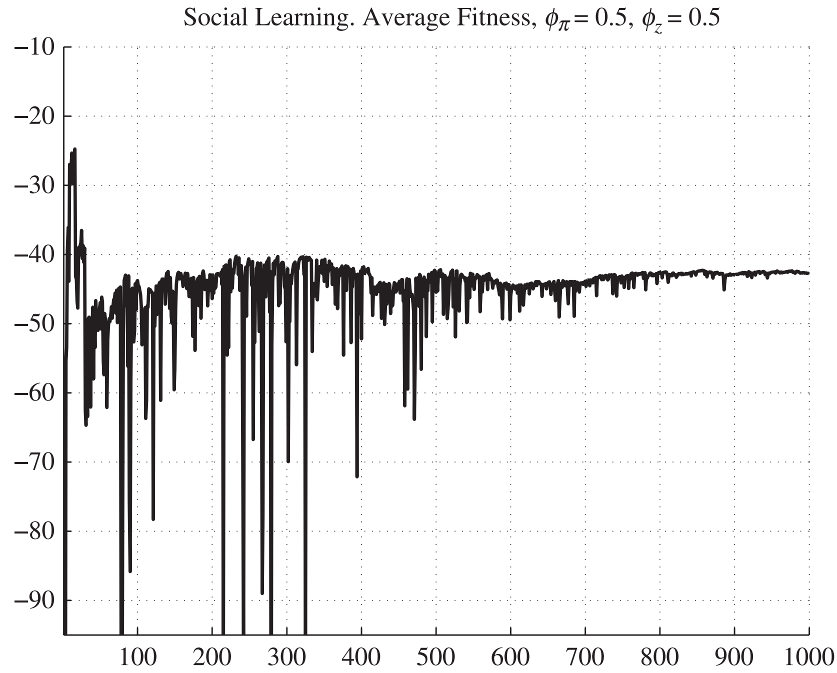

Fig. 7. *Fitness, Social Learning, Indeterminate and E-unstable Region:* $\phi_\pi = 0.5$, $\phi_z = 0.5$

social learning simulation start diverging away, their fitness goes down. At this point, rules that prescribe different values of the coefficients (if they are not already in the population, they are generated via crossover and mutation) will have higher fitness value and will start getting more copies, and will thus change the average value of the coefficients and the average fitness value which again increases. Figures 9 and 10 present time series of actual output gap and inflation deviation from target level and their forecasts in the respective simulations with constant gain least squares and social learning. We can see that as the coefficients deviate further away from their MSV values, the actual inflation deviation from target diverges away from zero. This is in contrast to what happens in the simulation with social learning where coefficients go to their MSV values and so actual values of output deviations and inflation deviations do not diverge.

We should also point out that how fitness is evaluated plays an important role. Each period agents evaluate how their forecasting model would have performed over all past periods, not only last period (21), (22) and (23). This is done so that agents are not swayed by one time shocks in our stochastic environment. The divergent models perform worse and therefore are eliminated through tournament selection.

The essential difference between the least squares type of algorithms and social learning is that the least squares updating scheme is not sensitive to the systematic forecasting error that the application of the algorithm results in the unstable regions of the parameter space. Unlike these algorithms, social learning is actively trying to increase the fitness of its rules, in other words, decrease their forecasting error. Anytime, a rule shows up that starts deviating from the neighbourhood of the MSV values, this has

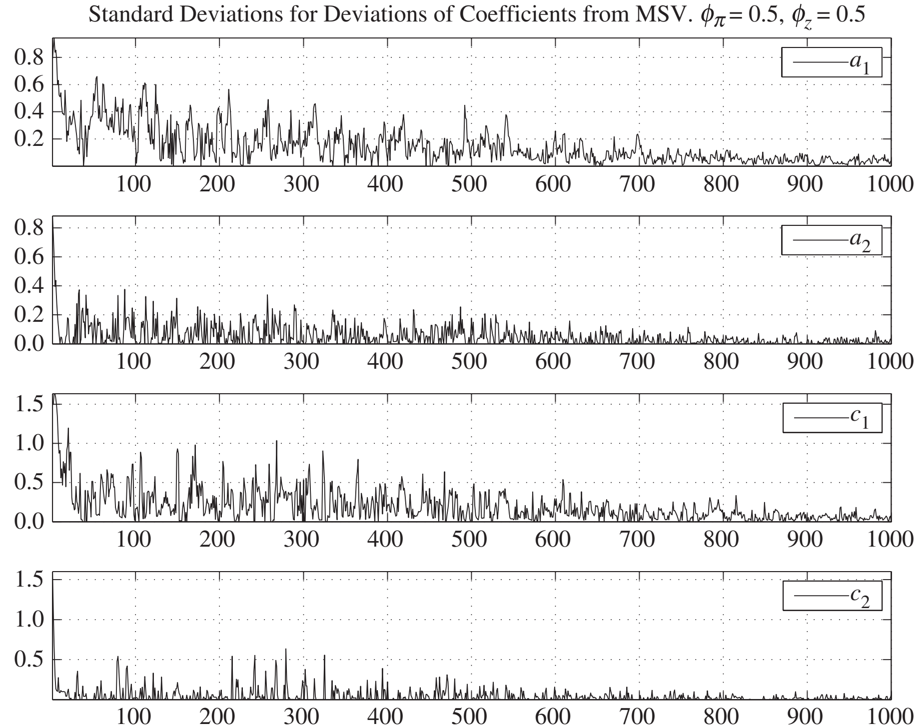

Fig. 8. *Standard Deviations of Coefficients, Social Learning, Indeterminate and E-unstable Region:* $\phi_\pi = 0.5, \ \phi_z = 0.5$

a negative impact on its fitness value, it becomes relatively less fit than the other rules and is eventually driven out of the population through tournament selection.

5. Learning of Sunspot Equilibria

Sunspot equilibria can exist in the indeterminate region of our model. In this Section, we investigate whether sunspot equilibria are learnable in our model. Sunspot equilibria can be represented in two forms – general form with uncorrelated sunspots and CF form with sunspot that follows an autoregressive process. The CF solution generalises finite state Markov sunspots. Evans and McGough (2005*a,b*) find that for certain models with certain policy rules CF sunspots can be learnable when general form sunspots are not learnable. In the model we use in this article, Evans and McGough (2005*b*) find that region of stability (instability) under learning is the same as the region of determinacy (indeterminacy)[20], and that all equilibria including CF sunspot solution are not stable under learning in the indeterminate region. We are going to study whether sunspot equilibria are learnable under social learning.

We proceed by presenting both types of sunspot equilibrium solutions. Our model can be written in this form:

[20] As is found by Bullard and Mitra (2002)

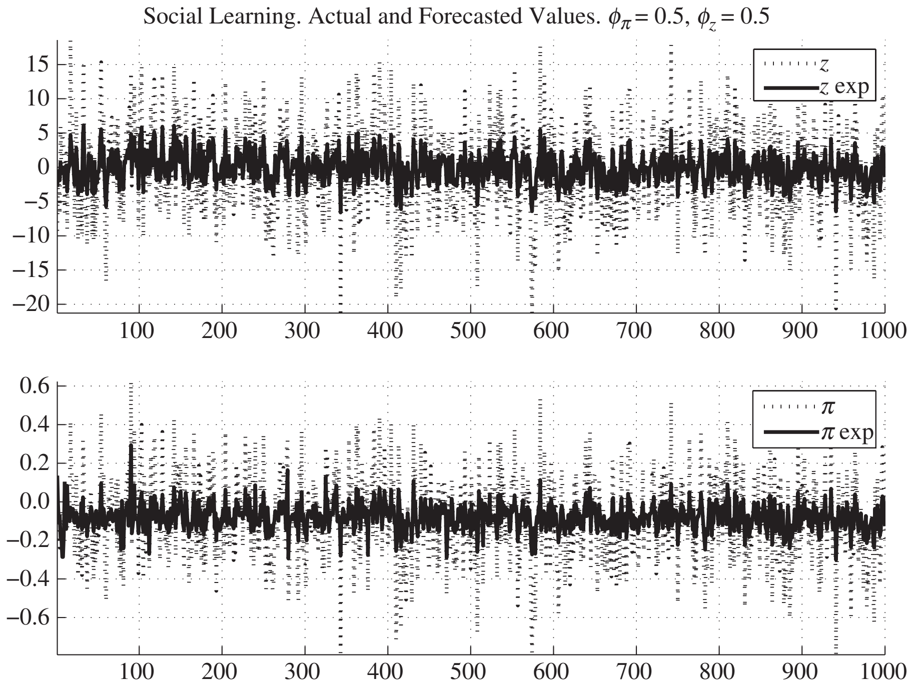

Fig. 9. *Actual and Forecasted Values, Constant Gain Learning, Indeterminate and E-unstable Region:* $\phi_\pi = 0.5,\ \phi_z = 0.5$

$$\boldsymbol{H}\boldsymbol{y}_t = \boldsymbol{F}E_t\boldsymbol{y}_{t+1} + \boldsymbol{M}r_t^n, \tag{32}$$

where

$$\boldsymbol{H} = \begin{bmatrix} 1+\sigma^{-1}\varphi_z & \sigma^{-1}\varphi_\pi \\ -\kappa & 1 \end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} 1 & \sigma^{-1} \\ 0 & \beta \end{bmatrix}, \quad \boldsymbol{M} = \begin{bmatrix} \sigma^{-1} \\ 0 \end{bmatrix}, \boldsymbol{y}_t = \begin{bmatrix} z_t \\ \pi_t \end{bmatrix}.$$

Let $\boldsymbol{\xi}_t = \boldsymbol{y}_t - E_t\boldsymbol{y}_t$, so that:

$$\begin{pmatrix} \boldsymbol{y}_t \\ r_t^n \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}^{-1}\boldsymbol{H} & -\boldsymbol{F}^{-1}\boldsymbol{M} \\ \boldsymbol{0}_{1x2} & \rho \end{pmatrix} \begin{pmatrix} \boldsymbol{y}_{t-1} \\ r_{t-1}^n \end{pmatrix} + \begin{pmatrix} \boldsymbol{F}^{-1} \\ \boldsymbol{0}_{1x2} \end{pmatrix} \xi_t + \begin{pmatrix} \boldsymbol{0}_{2x1} \\ 1 \end{pmatrix} \epsilon_t. \tag{33}$$

This can be rewritten as:

$$\hat{\boldsymbol{y}}_t = \hat{\boldsymbol{M}}\hat{\boldsymbol{y}}_{t-1} + \hat{\boldsymbol{w}}_t, \tag{34}$$

where $\hat{\boldsymbol{w}}_t = [w_{1,t} \quad w_{2,t}]'$ is a linear combination of the real interest rate shock ϵ_t and expectations shocks ξ_t. As ξ_t can be any martingale difference sequence, we can think of $\hat{w}_t$ as an arbitrary mds.

Matrix $\hat{\boldsymbol{M}}$ is diagonalised as $\hat{\boldsymbol{M}} = \boldsymbol{S}\boldsymbol{\Lambda}\boldsymbol{S}^{-1}$. The first two eigenvalues λ_1 and λ_2 in Λ are arranged in decreasing order. In our model, $|\lambda_1| > 1$ and $|\lambda_2| < 1$ for $\varphi_\pi = 0.5$, $\varphi_z \in [0.1,1.1]$, it is a case of order one indeterminacy. Both eigenvalues are outside the unit circle for all other parameter values which means that only fundamental REE exists.

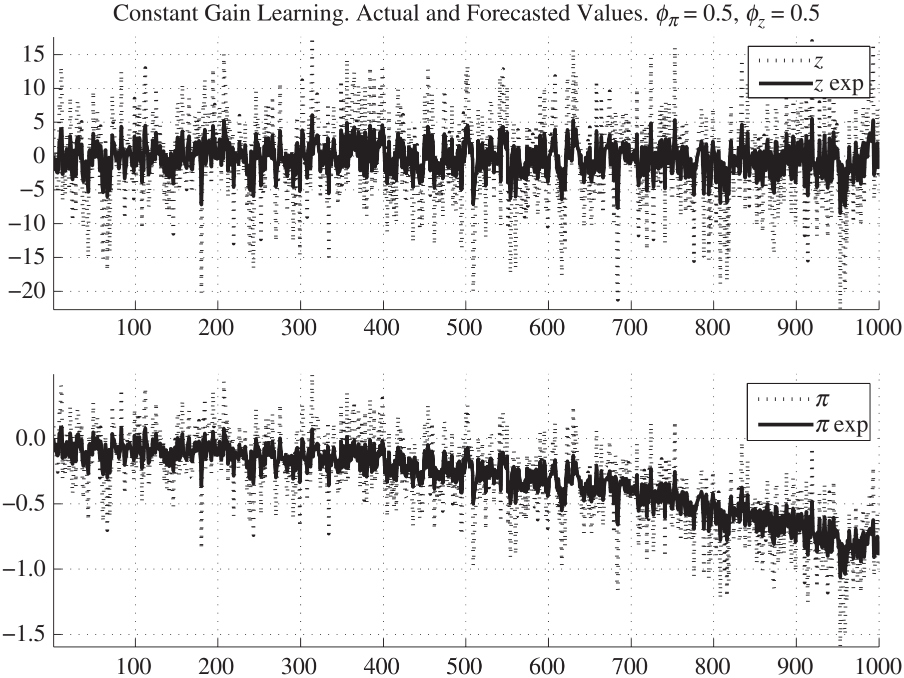

Fig. 10. *Actual and Forecasted Values, Social Learning, Indeterminate and E-unstable Region:* $\phi_\pi = 0.5,\ \phi_z = 0.5$

5.1. *General Form Representation*

The general form representation of a sunspot equilibrium in our model is:[21]

$$\boldsymbol{y}_t = \boldsymbol{a} + \boldsymbol{d}\boldsymbol{y}_{t-1} + \boldsymbol{c}r_t^n + \boldsymbol{f}r_{t-1}^n + \boldsymbol{g}w_{2,t}, \tag{35}$$

where $\boldsymbol{a} = \boldsymbol{0}$ and $w_{2,t}$ is an arbitrary one-dimensional martingale difference sequence (sunspot variable). $\boldsymbol{y}_t = [z_t \quad \pi_t]'$, $\boldsymbol{d}$ is a parameter matrix (2×2) and $\boldsymbol{c}$, $\boldsymbol{f}$ and $\boldsymbol{g}$ are parameter vectors (2×1). The general form representation nests the fundamental MSV solution in which $\boldsymbol{c} = \bar{\boldsymbol{c}}$ and the rest of coefficients are zero.

The social learning of general form solution is implemented in the same way as learning of the MSV solution. There are $N = 30$ agents indexed $i = 1, \ldots, N$. Agents know the specification of the sunspot equilibrium solution (35) but they do not know the values of the coefficients. Each agent learns about a vector of coefficients $(\boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{f}, \boldsymbol{g})$ in the sunspot solution. Thus, each agent i's PLM is given by:

$$\boldsymbol{y}_t = \boldsymbol{a}_{i,t} + \boldsymbol{d}_{i,t}\boldsymbol{y}_{t-1} + \boldsymbol{c}_{i,t}r_t^n + \boldsymbol{f}_{i,t}r_{t-1}^n + \boldsymbol{g}_{i,t}w_{2,t}, \tag{36}$$

where $\boldsymbol{a}_{i,t}, \boldsymbol{c}_{i,t}, \boldsymbol{f}_{i,t}, \boldsymbol{g}_{i,t}$ are (2×1) and $\boldsymbol{d}_{i,t}$ is (2×2).

[21] General form sunspot equilibrium solution is described in Evans and McGough (2005*b*)

The PLM (35) has the same form as the general form sunspot equilibrium solution (35), but the parameter values in PLM can be different from REE values. Coefficients are initialised around their respective sunspot equilibrium values similarly to initialisation in the case of learning about MSV solution.[22]

It has been shown in Honkapohja and Mitra (2004) that if agents' information set at period t includes values of endogenous variables in period t, the expectation of future endogenous variables $\hat{\mathrm{E}}_t \boldsymbol{y}_{t+1}$ is independent of the sunspot variable $w_{2,t}$ and therefore the ALM is independent of the sunspot variable because sunspot variable is martingale difference sequence $\mathrm{E}_t \boldsymbol{w}_{2,t+1} = 0$. When ALM is independent of sunspot variables, then the sunspot equilibrium solution is not learnable. Therefore, we assume that in period t agents cannot observe the values of endogenous variables in period t. They can observe endogenous variables up to period $t-1$. In period t, agents can observe the time t values of the exogenous variable and the sunspot variable.

In period t, each agent i makes forecasts of y_{t+1}:

$$\boldsymbol{y}^e_{i,t+1} = \boldsymbol{a}_{i,t} + \boldsymbol{d}_{i,t}\hat{\mathrm{E}}_t y_t + \boldsymbol{c}_{i,t}\hat{\mathrm{E}}_t r^n_{t+1} + \boldsymbol{f}_{i,t} r^n_t + \boldsymbol{g}_{i,t}\hat{\mathrm{E}}_t w_{2,t+1}, \tag{37}$$

where $\hat{\mathrm{E}}_t r^n_{t+1} = \rho r^n_t$, $\hat{\mathrm{E}}_t w_{2,t+1} = 0$ and forecast of $\boldsymbol{y}_t$, $\hat{\mathrm{E}}_t \boldsymbol{y}_t$, is determined by each agent i as:

$$\hat{\mathrm{E}}_t \boldsymbol{y}_t = \boldsymbol{a}_{i,t} + \boldsymbol{d}_{i,t}\boldsymbol{y}_{t-1} + \boldsymbol{c}_{i,t} r^n_t + \boldsymbol{f}_{i,t} r^n_{t-1} + \boldsymbol{g}_{i,t} w_{2,t}. \tag{38}$$

Using the individual agents' forecasts, the average forecasts of the output gap and deviation of inflation from target are computed as in (18) and (19). The actual values y_t are computed according to (20).

In each period, the pool of ideas is subject to learning operators – crossover, mutation and tournament selection – as described earlier. Agents evaluate the performance of their models based on fitness computed according to (21). An agent is characterised by a set of coefficients ($\boldsymbol{a}_{i,t}$, $\boldsymbol{d}_{i,t}$, $\boldsymbol{c}_{i,t}$, $\boldsymbol{f}_{i,t}$, $\boldsymbol{g}_{i,t}$) at each date t. The terms $z^f_{i,k}$ and $\pi^f_{i,k}$ in (21) are the forecasts of the output gap and the deviation of inflation from target that agent i could have computed in period k, if he had used the current, date t, set of coefficients ($\boldsymbol{a}_{i,t}$, $\boldsymbol{d}_{i,t}$, $\boldsymbol{c}_{i,t}$, $\boldsymbol{f}_{i,t}$, $\boldsymbol{g}_{i,t}$). These forecasts are computed using (37) and (38). When computing forecast in period k, agent uses information about endogenous variables up to period $k-1$ and exogenous variables and sunspot variable up to period k.

5.2. *Common Factor Representation*

The CF form of sunspot equilibrium in our model is:[23]

$$\boldsymbol{y}_t = \boldsymbol{a} + \boldsymbol{c} r^n_t + \boldsymbol{g}\eta_t, \tag{39}$$

where $\boldsymbol{a} = \boldsymbol{0}$, $\eta_t = \lambda_2 \eta_{t-1} + w_{2,t}$, where coefficient λ_2 is the second eigenvalue of matrix $\hat{\boldsymbol{M}}$, and $w_{2,t}$ is martingale difference sequence. $\boldsymbol{y}_t = [z_t \quad \pi_t]'$, $\boldsymbol{c}$ and $\boldsymbol{g}$ are vectors

[22] The initial values for the coefficients are each randomly generated from a normal distribution with mean equal to the respective sunspot equilibrium value. The standard deviation for coefficients c_1 and c_2 is equal to the largest of the absolute values of their sunspot values of these coefficients. We used initial standard deviation for the rest of the coefficients half as large as the standard deviation for the coefficients c.

[23] Common factor sunspot equilibrium solution is described in Evans and McGough (2005*b*)

(2×1). The CF representation nests the fundamental MSV solution in which $\boldsymbol{g} = \boldsymbol{0}$ and $\boldsymbol{c} = \bar{\boldsymbol{c}}$.

The social learning of CF representation of sunspot solution is implemented in the same way as learning of the MSV solution. There are $N = 30$ agents indexed $i = 1,\ldots,N$. Agents know the specification of the sunspot equilibrium solution (39), but they do not know the values of the coefficients. Agents learn about vector of coefficients $(\boldsymbol{a}, \boldsymbol{c}, \boldsymbol{g})$. Each agent i has the PLM:

$$\boldsymbol{y}_t = \boldsymbol{a}_{i,t} + \boldsymbol{c}_{i,t} r_t^n + \boldsymbol{g}_{i,t} \eta_t, \tag{40}$$

where $\boldsymbol{a}_{i,t}$,$\boldsymbol{c}_{i,t}$ and $\boldsymbol{g}_{i,t}$ are (2×1).

The PLM (40) has the same form as CF sunspot equilibrium solution (39), but the parameter values can be different from REE values. Coefficients are initialised around their respective sunspot equilibrium values similarly to initialisation in the case of learning about general form representation.[24]

In period t, each agent i makes forecasts of $\boldsymbol{y}_{t+1}$:

$$\boldsymbol{y}_{i,t+1}^e = \boldsymbol{a}_{i,t} + \boldsymbol{c}_{i,t} \hat{\mathrm{E}}_t r_{t+1}^n + \boldsymbol{g}_{i,t} \hat{\mathrm{E}}_t \eta_{t+1}, \tag{41}$$

where $\hat{\mathrm{E}}_t r_{t+1}^n = \rho r_t^n$ and $\hat{\mathrm{E}}_t \eta_{t+1} = \lambda_2 \eta_t$. As the sunspot variable η_t follows autoregressive process, the expectation $\hat{\mathrm{E}}_t \eta_{t+1}$ is not zero, and so the ALM includes sunspot variable even if agents can observe the endogenous variables up to period t. Therefore, we do not need to make the same assumption about information set that we had to make in case of general form representation.

The rest of the implementation is the same as for the general form representation. The average forecasts are determined as in (18) and (19). The actual values y_t are determined using (20).

In each period, a pool of agents' rules is subject to updating that consists of crossover, mutation and tournament selection – as described earlier. Agents evaluate performance of their rules based on their fitness values computed as (21).

An agent is characterised by a set of coefficients $(\boldsymbol{a}_{i,t}, \boldsymbol{c}_{i,t}, \boldsymbol{g}_{i,t})$ at each date t. The terms $z_{i,k}^f$ and $\pi_{i,k}^f$ are the forecasts of the output gap and the deviation of inflation from target that agent i could have computed in period k, if he had used the current, date t, set of coefficients $(\boldsymbol{a}_{i,t}, \boldsymbol{c}_{i,t}, \boldsymbol{g}_{i,t})$. These forecasts are computed according to (41).

5.3. *Computational Experiments*

We conduct a set of computational experiments to see whether the economy is able to co-ordinate on sunspot equilibria under social learning. We begin our simulations by generating initial history for the system at the sunspot equilibrium. When we study general form representation, the initial history is generated using the general form representation of sunspot equilibrium. When we study CF representation, the initial history is generated using CF sunspot equilibrium form. The initial history is 100 periods.

[24] The initial values for the coefficients are each randomly generated from a normal distribution with mean equal to the respective sunspot equilibrium value. The standard deviation for coefficients c_1 and c_2 is equal to the largest of the absolute values of their sunspot values of these coefficients. We used initial standard deviation for the rest of the coefficients half as large as the standard deviation for the coefficients c.

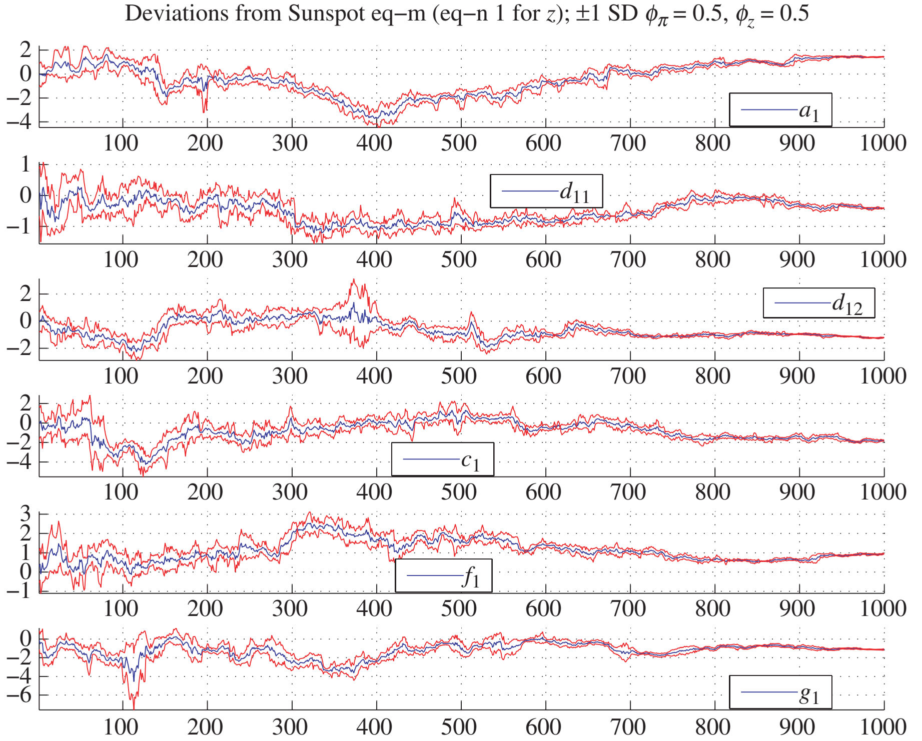

Fig. 11. *General Form Representation, Equation for z, Indeterminate and E-unstable Region:* $\phi_\pi = 0.5$, $\phi_z = 0.5$

Then we conduct simulations that last for 1,000 periods. We use the same parameter values as described earlier. The sunspot variable is generated as $w \sim N(0, \sigma_s)$, where $\sigma_s = 1$.[25]

We find that agents are not able to learn sunspot equilibrium solution in general form representation. A typical simulation for policy rule characterised by $\phi_\pi = 0.5$ and $\phi_z = 0.5$ is given in Figures 11 and 12. These figures show the time series of the deviations of each of the 12 coefficients from their sunspot equilibrium values averaged across all agents. Figure 11 shows these values in equation for output gap, z. Figure 12 shows these values in the equation for deviation of inflation from target level, π. These Figures also show ±1 standard deviation for each coefficient's deviation from sunspot equilibrium values, showing the extent of the dispersion in coefficients in period t among agents. From these Figures, we can see that most coefficients deviate from their sunspot equilibrium values. This means that the agents cannot learn sunspot equilibrium with social learning. For the same model, Evans and McGough (2005*b*)) show that there is no convergence in case of recursive least squares learning.

To further investigate these results, we performed 100 simulations for different policy rules (ϕ_π, ϕ_z) and collected in Table 7 the average deviations from sunspot

[25] Similar results were obtained for $\sigma_s = 0.001$.

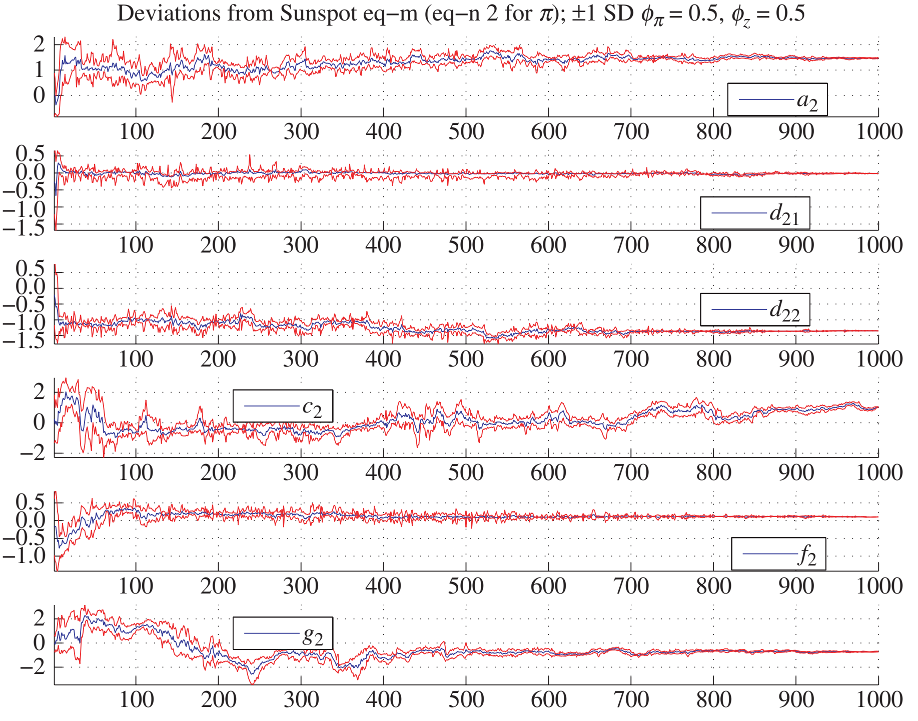

Fig. 12. *General Form Representation, Equation for* π*, Indeterminate and E-unstable Region:* $\phi_\pi = 0.5,\ \phi_z = 0.5$

equilibrium values computed similarly to data in Table 1.[26] Table 7 reports data for policy rules in indeterminate region. This Table shows that most coefficients exhibit large deviations from the sunspot equilibrium values. This means that the agents do not learn sunspot equilibrium in general form representation. Table 8 shows average values of coefficients g_1 and g_2 over 100 simulations. Non-zero values of these coefficients mean that agents take sunspot into account. But Table 7 shows that agents cannot learn the equilibrium values of most of the coefficients.

For a CF representation, a typical simulation for policy rule characterised by $\phi_\pi = 0.5$ and $\phi_z = 0.5$ is shown in Figure 13. This Figure shows time series of the deviations of each of 6 coefficients from their sunspot equilibrium values averaged across all agents ± 1 standard deviation. We can see that coefficients a_1 and a_2 do not go closely to their equilibrium values. Coefficients c_1 and c_2 go closely to their equilibrium values. Deviations of coefficients g_1 and g_2 from their sunspot equilibrium values are very large.

[26] We computed the average value of deviations of coefficients from the sunspot equilibrium values for each period. Then we computed average value of the average deviations during the last 100 periods of simulation. In addition, we collected data on percentage deviations from sunspot equilibrium during last 100 periods of the simulations. For each policy rule (φ_π, φ_z), we perform 100 simulations, collect the above described statistics for each simulation and report averages and standard deviations for each statistic over 100 simulations.

Table 7

Averages and Standard Deviations of Coefficient Deviations, General Form Sunspot Representation

	$\varphi_\pi = 0.5$											
φ_z	a_1	d_{11}	d_{12}	c_1	f_1	g_1	a_2	d_{21}	d_{22}	c_2	f_2	g_2
1.1	−0.0942	0.9562	1.9273	−0.6090	0.3200	0.2130	0.0291	3.1679	5.3576	−0.4133	0.2517	−0.5326
	(1.5723)	(2.2845)	(2.9136)	(2.6044)	(1.2446)	(1.7418)	(1.5928)	(3.7308)	(5.8357)	(2.1032)	(1.4067)	(1.4732)
%		62416.57	425.86	−73.54	−2128.70	51.55		93815.03	537.07	−1358.86	−759.71	−58.48
0.8	(0.3027	1.4435	2.3217	−0.6726	0.4520	−0.0001	−0.1569	4.1808	5.7569	−0.3918	0.3237	−0.4230
	(2.0607)	(2.5852)	(3.7237)	(3.8455)	(1.6123)	(2.332)	(1.9918)	(4.7224)	(7.0545)	(2.4719)	(1.5763)	(1.9591)
%		51001.30	378.42	−61.05	−1617.46	−0.03		91328.48	580.17	−968.28	−716.21	−49.74
0.5	0.0072	2.4719	2.6457	−0.7110	0.2869	0.6928	0.1719	6.2456	6.8947	−1.1715	0.1521	−0.5953
	(3.0046)	(4.6493)	(5.3942)	(5.8578)	(2.502)	(3.0647)	(2.8616)	(7.3662)	(10.0197)	(4.9908)	(2.7737)	(2.9866)
%		36001.79	278.64	−43.20	−417.60	99.55		88157.45	703.72	−1938.06	−214.64	−82.91
0.2	0.2477	6.1196	2.9331	−1.4629	0.2170	1.3245	−0.0137	8.5229	8.2788	−0.6814	0.2230	−0.9555
	(7.1649)	(11.1359)	(9.999)	(14.0602)	(6.3803)	(7.1033)	(7.7260)	(17.2887)	(19.3174)	(11.8777)	(6.9291)	(7.1366)
%		18231.61	143.94	−44.99	−61.58	145.41		56056.68	896.93	−570.62	−139.69	−231.58

Table 8

Average Coefficients g_1 and g_2, General Form (Standard Deviations in parenthesis)

	$\varphi_\pi = 0.5$	
φ_z	g_1	g_2
1.1	0.6261	0.3781
	(1.7418)	(1.4732)
0.8	0.5257	0.4275
	(2.332)	(1.9591)
0.5	1.3888	0.1228
	(3.0647)	(2.9866)
0.2	2.2354	−0.5429
	(7.1033)	(7.1366)

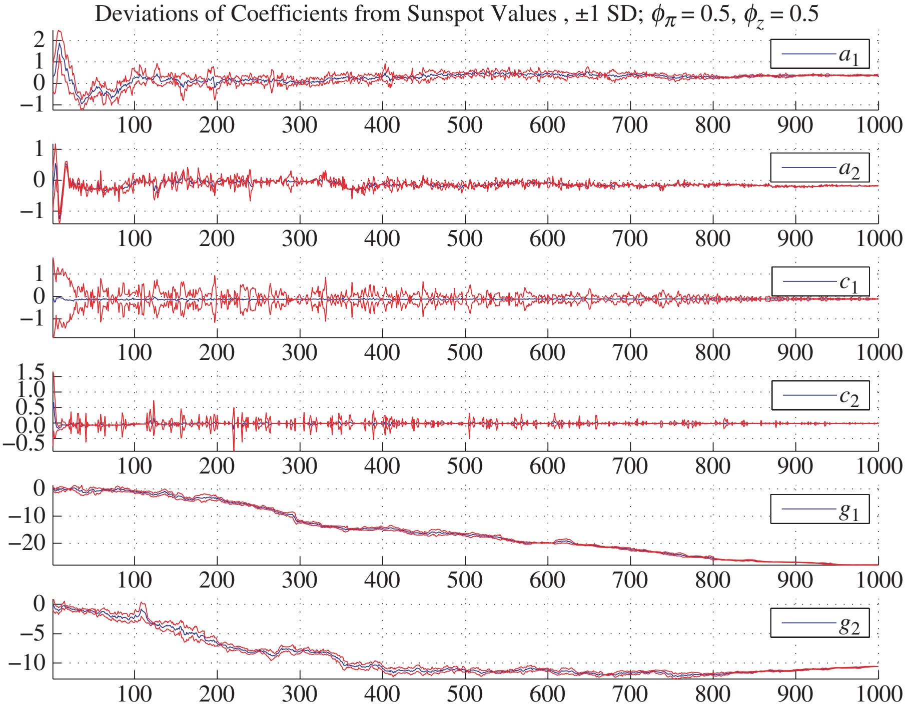

Fig. 13. *Common Factor Representation, Indeterminate and E-unstable Region:* $\phi_\pi = 0.5$, $\phi_z = 0.5$

To further investigate these results, we performed 100 simulations for different policy rules (ϕ_π,ϕ_z) and collected in Table 9 the same data as in Table 7. Table 9 shows that coefficients a_1,a_2,c_1,c_2 go to their equilibrium values very closely[27] but coefficients g_1 and g_2 do not go to their sunspot equilibrium values. Table 10 shows the average values of coefficients g_1 and g_2 during last 100 periods of simulation, averaged over 100 simulations. We can see that coefficients g_1 and g_2 do not go to 0 that means that the

[27] Sunspot equilibrium values of these coefficients are equal to their MSV equilibrium values.

Table 9

Average Coefficient Deviations, Common Factor Representation (Standard Deviations in parenthesis)

	$\varphi_\pi = 0.5$					
φ_z	a_1	c_1	g_1	a_2	c_2	g_2
1.1	−0.0371	−0.0223	−0.0363	−0.0770	−0.0038	−0.0795
	(0.3771)	(0.0716)	(0.0936)	(0.5883)	(0.0371)	(0.2084)
%		−2.69	−8.78		−12.39	−8.73
0.8	−0.0657	−0.0319	−0.0586	−0.0909	−0.0067	−0.0986
	(0.4878)	(0.0963)	(0.1682)	(0.6436)	(0.0372)	(0.2737)
%		−2.89	−11.15		−16.66	−11.60
0.5	−0.0246	−0.0537	−0.0823	−0.0322	−0.0120	−0.0894
	(0.8331)	(0.1492)	(0.3499)	(0.7950)	(0.0386)	(0.3638)
%		−3.26	−11.82		−19.82	−12.45
0.2	0.2188	−0.1416	−0.1360	0.1013	−0.0174	−0.0720
	(3.7078)	(0.3257)	(1.5336)	(1.7395)	(0.0540)	(0.7050)
%		−4.36	−14.93		−14.61	−17.46

Table 10

Average Coefficients g_1 and g_2, Common Factor Representation (Standard Deviations in parenthesis)

	$\varphi_\pi = 0.5$	
φ_z	g_1	g_2
1.1	0.3769	0.8312
	(0.0936)	(0.2084)
0.8	0.4673	0.7519
	(0.1682)	(0.2737)
0.5	0.6137	0.6287
	(0.3499)	(0.3638)
0.2	0.7749	0.3406
	(1.5336)	(0.705)

agents take sunspot variable into consideration. Looking at the percentage deviations, we can see that deviations are in the range between 2.69% and 4.36% for c_1, between 12.39% and 19.82% for c_2 and between 8.73% and 17.46% for g_1 and g_2.

These Figures and Tables can be interpreted as agents co-ordinate on a sunspot of their own, in a 'learning' equilibrium. This is reflected in non-zero values of coefficients g_1 and g_2 that are different from their sunspot equilibrium values. As Evans and McGough (2005*a*) point out '… g appears to converge to some number not equal to zero, suggesting that agents in this economy will indeed learn that there is a dependence on a sunspot variable' (Evans and McGough, 2005*a*, p. 618). Our result is similar to that of Evans and McGough (2005*a,b*): although our agents cannot learn sunspot equilibrium in general form, they are able to learn it in CF form.

Table 11
Average Fitness

	$\varphi_\pi = 0.5$		
φ_z	Fit average	Fit CF ss	Fit CF eq-m
1.1	−6810.6062 (7345.8423)	−1581.8217 (3118.9041)	−101.8105 (0.8954)
0.8	−924.2446 (709.8466)	−887.1921 (1571.165)	−98.7563 (0.7717)
0.5	−110.2422 (64.5408)	−672.9065 (935.6405)	−99.9892 (0.5156)
0.2	−1126.2910 (1530.2947)	−1069.5502 (1105.0479)	−187.0296 (0.0506)

5.4. *Computation of Fitness*

We have computed fitness in the environment with social learning about CF representation. We have evaluated the average fitness of learning agents and the fitness of rule with CF sunspot equilibrium values in the simulation with social learning.[28] The statistics are presented in columns 2 and 3 of Table 11. We have also evaluated fitness of rule with CF equilibrium values in the environment generated by CF sunspot equilibrium solution. The data are presented in column 4 of Table 11.

We performed these calculations to see how learning agents are affected by creating their own 'sunspot' variable. As we can see from Table 11, the average fitness of learning agents is substantially lower than the fitness of rule with CF equilibrium values in the learning environment for most of the parameter combinations. It is also much lower than fitness of CF equilibrium rule in the environment generated by CF equilibrium. These results mean that learning agents cannot achieve the performance as high as that of CF equilibrium rule when they create their own sunspot.

6. Conclusion

A key finding in the literature on learning in New Keynesian models of monetary policy is that nominal interest rate feedback policies which are too close to an interest rate peg tend to be associated with indeterminacy and instability in the recursive learning dynamics. The policy maker must react sufficiently aggressively to economic developments to assure determinacy of rational expectations equilibrium and expectational stability of that equilibrium. This has been promoted as an important reason to discard policy rules which are insufficiently aggressive,[29] and this idea has gained widespread acceptance in monetary policy discussions.

We have investigated whether this result is robust to the substitution of an evolutionary learning dynamic for the recursive learning dynamic. Our main finding is that

[28] The fitness values presented here are computed based on the average values over the last 100 periods of each simulation that are then averaged over 100 simulations for each set of the parameter values.
[29] For instance, Woodford (2003).

the evolutionary learning dynamic does not put a premium on policy rules which obey the Taylor Principle. Instead, evolutionary learning converges in simulation to a small neighbourhood of the MSV solution whether or not the policy maker obeys that principle.

When the Taylor Principle is violated, equilibrium is indeterminate. It is well known that sunspot equilibria exist in a neighbourhood of an indeterminate rational expectations equilibrium. This is another important reason why insufficiently aggressive policy rules may be considered poor policy. In the recursive learning literature, it has generally been difficult to obtain expectational stability of sunspot equilibria.[30]

We thus extend our analysis to examine the stability of sunspot equilibria under our evolutionary learning dynamic and study two different representations of sunspot equilibria, general form and CF. Both representations increase the number of coefficients that our agents are learning about.[31] Our agents fail to co-ordinate on a sunspot equilibrium in general form representation, however, they are able to learn sunspot solution in CF form.

If agents start with MSV REE specification, they are able to learn its equilibrium values whether or not Taylor Principle holds. If agents start with sunspot CF equilibrium form, they are able to co-ordinate on sunspot solution. This provides grounds for caution regarding interest rate rules not satisfying Taylor Principle because CF sunspot solutions can be learnable in the indeterminate region.

Simon Fraser University
Federal Reserve Bank of St. Louis
Portland State University

Submitted: 24 January 2008
Accepted: 3 October 2011

References

Arifovic, J. (1994). 'Genetic algorithm learning and the cobweb model', *Journal of Economic Dynamics and Control*, vol. 18, pp. 3–28.

Arifovic, J. (1995). 'Genetic algorithms and inflationary economies', *Journal of Monetary Economics*, vol. 36, pp. 219–43.

Arifovic, J. (1996). 'The behavior of the exchange rate in the genetic algorithm and experimental economics', *Journal of Political Economy*, vol. 104, pp. 510–41.

Arifovic, J. (2000). 'Evolutionary algorithms in macroeconomic models', *Macroeconomic Dynamics*, vol. 4(3), pp. 373–414.

Arifovic, J., Bullard, J. and Duffy, J. (1997). 'The transition from stagnation to growth: an adaptive learning approach', *Journal of Economic Growth*, vol. 2, pp. 185–209.

Arifovic, J. and Ledyard, J. (2010). 'A behavioral model for mechanism design: individual evolutionary learning', *Journal of Economic Behavior and Organization*, vol. 78, pp. 374–95.

Arifovic, J. and Maschek, M. (2006). 'Revisiting individual evolutionary learning in the cobweb model – an illustration of the virtual spite effect', *Computational Economics*, vol. 28, pp. 333–54.

Back, T., Fogel, D.B. and Michalewicz, Z. (2000). *Evolutionary Computation 1 Basic Algorithms and Operators*, Bristol and Philadelphia: Institute of Physics Publishing.

Blanchard, O. and Kahn, C. (1980). 'The solution of linear difference models under rational expectations', *Econometrica*, vol. 48(5), pp. 1305–11.

[30] For instance, Honkapohja and Mitra (2004).
[31] It is equal to 12 for the general form representation and 6 for the CF representation.

Branch, W. and Evans, G. (2006). 'Intrinsic heterogeneity in expectation formation', *Journal of Economic Theory*, vol. 127(1), pp. 264–95.

Branch, W. and McGough, B. (2009). 'A New Keynesian model with heterogeneous expectations', *Journal of Economic Dynamics and Control*, vol. 33(5), pp. 1036–51.

Bullard, J. (2006). 'The learnability criterion and monetary policy', *Federal Reserve Bank of St. Louis Review*, vol. 88(3), pp. 203–17.

Bullard, J. and Duffy, J. (1998). 'Using genetic algorithms to model the evolution of heterogenous beliefs', *Computational Economics*, vol. 13(1), pp. 41–60.

Bullard, J. and Duffy, J. (2001). 'Learning and excess volatility', *Macroeconomic Dynamics*, vol. 5, pp. 272–302.

Bullard, J. and Duffy, J. (2004). 'Learning and structural change in macroeconomic data', Federal Reserve Bank of St. Louis Working Paper No. 2004-016A.

Bullard, J. and Mitra, K. (2002). 'Learning about monetary policy rules', *Journal of Monetary Economics*, vol. 49(6), pp. 1105–29.

Dawid, H. (1996). 'Learning of cycles and sunspot equilibria by genetic algorithms', *Journal of Evolutionary Economics*, vol. 6(4), pp. 361–73.

Dawid, H. (1999). *Adaptive Learning by Genetic Algorithms: Analytical Results and Applications to Economic Models*, Berlin: Springer.

Dawid, H. and Kopel, M. (1998). 'The appropriate design of a genetic algorithm in economic applications exemplified by a model of the cobweb type', *Journal of Evolutionary Economics*, vol. 8, pp. 297–315.

Evans, G. and Honkapohja, S. (2001). *Expectations and Learning in Macroeconomics*, Princeton: Princeton University Press.

Evans, G. and Honkapohja, S. (2003). 'Expectations and the stability problem for optimal monetary policies', *Review of Economic Studies*, vol. 70(4), pp. 807–24.

Evans, G. and McGough, B. (2005*a*). 'Stable sunspot solutions in models with predetermined variables', *Journal of Economic Dynamics and Control*, vol. 29, pp. 601–25.

Evans, G. and McGough, B. (2005*b*). 'Monetary policy, indeterminacy and learning', *Journal of Economic Dynamics and Control*, vol. 29, pp. 1809–40.

Franke, R. (1998). 'Coevolution and stable adjustment in the cobweb model', *Journal of Evolutionary Economics*, vol. 8, pp. 383–406.

Giannitsarou, C. (2003). 'Heterogeneous learning', *Review of Economic Dynamics*, vol. 6(4), pp. 885–906.

Gintis, H. (2007). 'The dynamics of general equilibrium', Economic Journal, vol. 117, pp. 1280–309.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimizations, and Machine Learning*, Reading: Addison-Wesley Pub.Co.

Haruvy, E., Roth, A.E. and Ünver, U.M. (2006). 'The dynamics of law clerk matching: an experimental and computational investigation of proposals for reform of the market', *Journal of Economic Dynamics and Control*, vol. 30, pp. 457–86.

Herrera, F., Lozano, M. and Verdegay, J.L. (1998). 'Tackling real-coded genetic algorithms: operators and tools for behavioural analysis', *Artificial Intelligence Review*, vol. 12, pp. 265–319.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press.

Honkapohja, S. and Mitra, K. (2004). 'Are non-fundamental equilibria learnable in models of monetary policy?', *Journal of Monetary Economics*, vol. 51(8), pp. 1743–70.

Honkapohja, S. and Mitra, K. (2005). 'Performance of monetary policy with internal central bank forecasting', *Journal of Economic Dynamics and Control*, vol. 29, pp. 627–58.

Honkapohja, S. and Mitra, K. (2006). 'Learning stability in economies with heterogeneous agents', *Review of Economic Dynamics*, vol. 9(2), pp. 284–309.

LeBaron, B. (2006). 'Agent-based computational finance', in (L.Tesfatsion and K. Judd, eds.), *Handbook of Computational Economics*, pp. 1187–232, Amsterdam: North Holland.

LeBaron, B., Arthur, W. and Palmer, R. (1999). 'Time series properties of an artificial stock market', *Journal of Economic Dynamics and Control*, vol. 23, pp. 1487–516.

Lux, T. and Marchesi, M. (2000). 'Volatility clustering in financial markets: a micro-simulation of interacting agents', *International Journal of Theoretical and Applied Finance*, vol. 3, pp. 675–702.

Lux, T. and Schornstein, S. (2002). 'Genetic learning as an explanation of stylized facts of foreign exchange markets', *Journal of Mathematical Economics*, vol. 41(1-2), pp. 169–96.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*, Berlin and Heidelberg: Springer-Verlag.

Negroni, G. (2003). 'Adaptive expectations coordination in an economy with heterogeneous agents', *Journal of Economic Dynamics and Control*, vol. 28(1), pp. 117–40.

Schwefel, H.P. (2000). 'Advantages (and disadvantages) of evolutionary computation over other approaches', in (T. Back, D.B. Fogel and Z.Michalewicz, eds.), *Evolutionary Computation 1: Basic Algorithms and Operators*, pp. 20–2, Bristol and Philadelphia: Institute of Physics Publishing.

Sensfuß, F., Ragwitz, M., Massimo Genoese, M. and Mst, D. (2007). 'Agent-based simulation of electricity markets: a literature review', Working Paper, Fraunhofer Institute Systems and Innovation Research.

Vriend, N.J. (2000). 'An illustration of the essential difference between individual and social learning, and its consequences for computational analyses', *Journal of Economic Dynamics and Control*, vol. 24, pp. 1–19.

Weidlich, A. and Veit, D. (2008). 'A critical survey of agent-based wholesale electricity market models', *Energy Economics*, vol. 30(4), pp. 1728–59.

Woodford, M. (2001). 'The Taylor rule and optimal monetary policy', *American Economic Review*, vol. 91(2), pp. 232–7.

Woodford, M. (2003). *Interest and Prices: Foundations of a Theory of Monetary Policy*, Princeton: Princeton University Press.