Computational Intelligence in Agent-Based Computational Economics

Shu-Heng Chen

AI-ECON Research Center, Department of Economics, National Chengchi University, Taipei, Taiwan 116, chchen@nccu.edu.tw

1 Introduction

1.1 What is *Agent-Based Computational Economics* (ACE)?

Agent-based computational economics is the study of economics using *agent-based modeling and simulation*, which, according to [21], is the third way, in addition to deduction and induction, to undertake social sciences. An agent-based model is a model comprising *autonomous agents* placed in an interactive environment (society) or social network. Simulating this model via computers is probably the most practical way to visualize economic dynamics.

An autonomous agent is one which is able to behave (think, learn, adapt, make strategic plans) with a set of specifications and rules which are given initially; they are fixed and require no further intervention. The necessity for using autonomous agents in agent-based computational economics – or, more broadly, agent-based social sciences – is still an issue open for discussion. We make no attempt here to give a full account of the development of this issue – this would deserve a Chapter on its own. For a brief account, the use of autonomous agents is, in one way or the other, connected to the notion of *bounded rationality*, popularized by Herbert Simon [138].

In order to build autonomous agents, agent-based computational economists need to employ existing algorithms or develop new algorithms which can enable agents to behave with a degree of autonomy. Sections 1.2, 2 and 3 of the chapter will give a thorough review of the algorithmic foundations of ACE. We also introduce here the field known as *computational intelligence* (CI) and its relevance to economics. Section 4 then reviews the use of computational intelligence in agent-based economic and financial models. Section 5 gives some general remarks on these applications, as well as pointing to future directions. This is followed by concluding remarks in Sect. 6.

S.-H. Chen: *Computational Intelligence in Agent-Based Computational Economics*, Studies in Computational Intelligence (SCI) **115**, 517–594 (2008)
www.springerlink.com

1.2 Algorithmic Foundations of ACE

The purpose of the next few Sections is to address a very important attribute of *autonomous agents*, this being their capability to adapt to a changing environment. The idea is to equip the artificial agents with some in-built algorithms so that they are able to develop some degree of sophisticated cognitive behavior; in particular, they are able to *learn from the past*, and hence are able to anticipate the future, and develop strategic behavior accordingly. The algorithms which can help our artificial agents achieve the above goal are initiated from many different fields and hence are interdisciplinary. Recently, they have been addressed together in the field known as *computational intelligence (CI)*. Therefore, the next few Sections may be read as an introduction to CI from the perspective of agent-based computational economics.

The aim of this Chapter is to review a number of important developments in computational intelligence, including artificial neural networks (Sect. 2) and evolutionary computation (Sect. 3). While these tools have been introduced to economists on numerous other occasions – for example, quantitative finance – we have a different motivation for studying them. Mainly, these two major CI tools allow us to discuss a number of crucial mental activities, such as attention control, memory, and pattern discovery. Therefore, even though our brief review will go through some important quantitative applications, we should remind readers at different places that our scope is broader.

Section 2 describes a number of different neural network models, which help us to understand how some algorithms, associated with the artificial brain, are able to conduct data compression, redundancy removal, classification, and forecasting. Let us be more specific with some of these. An important cognitive task for human agents is that, under some degree of survival pressure (or incentives), they are able to perform correct classification and react upon it properly. A simple example is the salesman who needs to identify those consumers who are willing to pay a high price for a specific new product, and to distinguish them from general buyers. A family of neural networks, also known as *supervised learning* (Sect. 2.1, 2.2, and 2.5), are able to equip agents with this capability.

Prior to classification, one more fundamental cognitive task is *concept formation* – in other words, to extract useful concepts from observations. Then, based on these concepts, new observations can be classified so as to facilitate decision-making. A typical example would be a stock trader who needs to recognize some special charts to make his market timing decisions. A family of neural networks, known as *unsupervised learning* (Sect. 2.6), can help agents to acquire this kind of cognitive capability.

Sometimes, it is hard to form concepts. In this case, one may directly deal with *cases*, and make decisions based on the similarity of cases. Sections 2.7 and 2.8 are devoted to the literature on *lazy learning* – that is, learning by simply memorizing experiences, with little effort to develop generalized concepts on top of these experiences.

The third important cognitive task concerns the efficient use of limited brain space. This has something to do with data compression or redundancy removal. Section 2.4 introduces a network which can perform this task. In addition, Sect. 2.3 describes a device to reduce data storage space by building in loops in the 'brain'.

The above three cognitive tasks do not involve social interaction. They mainly describe how an individual learns from his own experience without interacting with other individuals' experiences. The latter case is referred to as *social learning* or *population learning* in the literature. Imitation (reproduction) is the clearest example of social learning: agents simply follow the behavior rules of whomever they consider the most suitable. Nonetheless, imitation is not enough to cover more complex patterns of social learning, such as innovations of using inspiration from others. Through evolutionary computation (Sect. 3), both forms (imitation and innovation) of learning with social interactions are operated with the familiar survival-of-the-fittest principle.[1] Genetic programming (GP) is a one kind of evolutionary computation. It differs from others in the sense that it gives us much more expressive power to observe changes.

2 Artificial Neural Networks

Among CI tools, the artificial neural network (ANN) is the most widely acceptable tool for economists and finance people, even though its history is much shorter than that of fuzzy logic so far as the application to economics and finance is concerned. The earliest application of ANNs was [156]. Since then we have witnessed an exponential growth in the number of applications. ANN is probably the only CI tool which drew serious econometricians' attention and on which a lot of theoretical studies have been done. Both [131] and [157] gave a rigorous mathematical/statistical treatment of ANNs, and hence have established ANNs with a sound foundation in the econometrics field. Nowadays, ANNs have already become an integral part of textbooks in econometrics, and even moreso in financial econometrics and financial time-series. A great number of textbooks or volumes especially edited for economists and finance people are available, for example, [23, 24, 84, 130, 136, 147, 163], to name a few. Its significance to finance people can also be seen from the establishment of the *Neurove$t* journal (now *Computational Intelligence in Finance*) in 1993.

It has been shown in a great number of studies that artificial neural nets, as representative of a more general class of nonlinear models, can outperform

[1] Evolutionary computation, in a sense, is a kind of 'bio-sociology'.

many linear models, and can sometimes also outperform some other nonlinear models.[2]

Three classes of artificial neural nets have been most frequently used in economics and finance. These are *multilayer perceptrons*, *radial basis neural networks*, and *recurrent neural networks*. The first two classes will be introduced in Sects. 2.1 and 2.2, whereas the last one is introduced in 2.3.

2.1 Multilayer Perceptron Neural Networks

Let us consider the following general issue. We observe a time-series of an economic or financial variable, such as the foreign exchange rate, $\{x_t\}$. We are interested in knowing its future values, $x_{t+1}, x_{t+2}, ...$ For that purpose, we need to search for a function relation $f(\ )$, such that when a vector $\mathbf{x}_t$ is input into the function, a prediction on $x_{t+1}, ...$ can be made. The question then is how to construct such a function. Tools included in this Chapter provide two directions in which to work, distinguishable by different modeling philosophies. The first one is based on the *universal modeling approach*, and the second one is based on the *local modeling approach*. Alternatively, we can say that the first one aims to build the function in the *time domain*, whereas the second works in the *feature* or *trajectory domain*.[3] We shall start with the first approach, and the canonical artificial neural network (ANN) can be considered to be a representative of this paradigm.

The reason why economists can embrace the ANN without any difficulty is due to the fact that ANN can be regarded as a generalization of their already familiar time-series model, *ARMA* (autoregressive moving-average) model. Formally, an ARMA(p,q) model is described as follows:

$$\Phi(L)x_t = \Theta(L)\epsilon_t \tag{1}$$

where $\Phi(L)$ and $\Theta(L)$ are polynomials of order p and q,

$$\Phi(L) = 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p \tag{2}$$

$$\Theta(L) = 1 - \theta_1 L - \theta_2 L^2 - \cdots - \theta_q L^q \tag{3}$$

$\{\epsilon_t\}$ is white noise, and L is the lag operator.

ANNs can be regarded as a non-linear generalization of these ARMA processes. In fact, more concretely, *multilayer perceptron (MLP)neural networks* are nonlinear generalizations of the so-called autoregressive process,

$$x_t = f(x_{t-1}, \ldots, x_{t-p}) + \epsilon_t \tag{4}$$

[2] This is not an appropriate place to provide a long list, but interested readers can find some examples from [7, 77, 90, 137, 153, 154] and [158].

[3] There are alternate names for the local modeling approach, for example *'guarded experts'* – see [18].

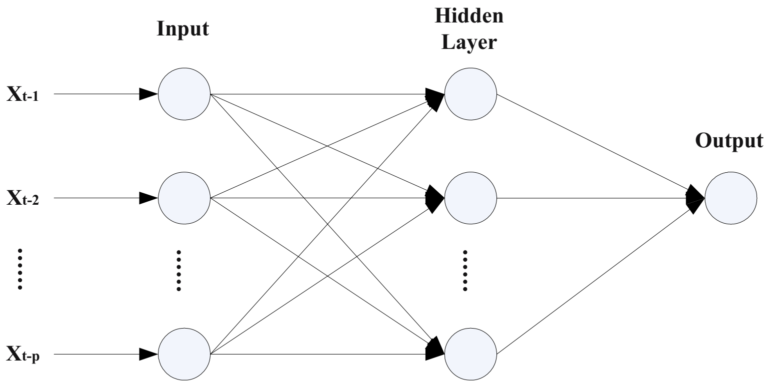

Fig. 1. The multilayer perceptron neural network of a non-linear AR process

whereas *recurrent neural networks* are non-linear generalizations of the ARMA processes,

$$x_t = f(x_{t-1}, \ldots, x_{t-p}, \epsilon_{t-1}, ...\epsilon_{t-q}) + \epsilon_t \tag{5}$$

In terms of a multilayer perceptron neural network, Eqn. (4) can then be represented as:

$$x_t = h_2(w_0 + \sum_{j=1}^{l} w_j h_1(w_{0j} + \sum_{i=1}^{p} w_{ij} x_{t-i})) + \epsilon_t \tag{6}$$

Hence Eqn. (6) is a three-layer neural net (Fig. 1). The input layer has p inputs: $x_{t-1}, ...x_{t-p}$. The hidden layer has l hidden nodes, and there is a single output for the output layer $\hat{x}_t$. Layers are fully connected by *weights*; w_{ij} is the weight assigned to the ith input for the jth node in the hidden layer, whereas w_j is the weight assigned to the jth node (in the hidden layer) for the output; w_0 and w_{0j} are constants, also called *biases*; h_1 and h_2 are *transfer functions*.

There is a rich choice of transfer functions. According to [64], a multilayer perceptron network with any *Tauber-Wiener functions* as transfer function of the hidden units can be qualified as a *universal approximator*. Also, a necessary and sufficient condition for being a Tauber-Wiener function is that *it is non-polynomial*. In practice, a differentiable transfer function is desirable. Commonly used transfer functions for multilayer perceptron networks are the *sigmoid function*,

$$h_s(x) = \frac{1}{1 + e^{-x}} \tag{7}$$

and *hyperbolic tangent function*,

$$h_t(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{8}$$

Clearly, $0 < h_s(x) < 1$, and $-1 < h_t(x) < 1$.

2.2 Radial Basis Network

Next to the multilayer perceptron neural network is the *radial basis network* (RBN), which is also popularly used in economics and finance. Radial basis function (RBF) networks are basically a feedforward neural networks with a *single hidden layer*,

$$f(x) = \sum_{i}^{k} w_i \varphi(\|x - c_i\|), \tag{9}$$

where $\varphi(\;)$ is a *radial basis function*, c_i is the ith center, and k is the number of the center. Both w_i, c_i and k are determined by the data set of x. Typical choices of radial basis functions are:

- the *thin-plate-spline function*,

$$\varphi(x) = x^2 \log^x \tag{10}$$

- the *Gaussian function*,

$$\varphi(x) = \exp(-\frac{x^2}{\beta}) \tag{11}$$

- the *multi-quadratic function*,

$$\varphi(x) = (x^2 + \beta^2)^{\frac{1}{2}} \tag{12}$$

- the *inverse multi-quadratic function*,

$$\varphi(x) = \frac{1}{(x^2 + \beta^2)^{\frac{1}{2}}} \tag{13}$$

Theoretical investigation and practical results seem to show that the choice of radial basis function is not crucial to the performance of the RBF network.

It has been proved that the RBF network can indeed approximate arbitrarily well any continuous function if a sufficient number of radial-basis function units are given (the network structure is large enough), and the network parameters are carefully chosen. RBN also has the best approximation property in the sense of having the minimum distance from any given function under approximation.

2.3 Recurrent Neural Networks

In Sect. 2.1, we discussed the relation between time series models and artificial neural networks. Information transmission in the usual multilayer perceptron neural net is *feedforward* in the sense that information is transmitted *forward* from the input layer to the output layer, via all hidden layers in between, as shown in Fig. 1; transmission in the reverse direction between any two layers is not allowed.

This specific architecture makes the multilayer perceptron neural net unable to deal with the moving-average series, $MA(q)$, effectively. To see this, consider the following $MA(1)$ series:

$$x_t = \epsilon_t - \theta_1 \epsilon_{t-1} \tag{14}$$

It is well-known that if $|\theta_1| < 1$, then the above MA(1) series can also be written as an AR(∞) series.

$$x_t = -\sum_{i=1}^{\infty} \theta^i x_{t-i} + \epsilon_t \tag{15}$$

In using the multilayer perceptron neural network to represent Eqn. (15), one needs to have an input layer with an infinite number of neurons (*infinite memory of the past*), namely, $x_{t-1}, x_{t-2}, ...$, which is impossible in practice. Although from the viewpoint of approximation, an exact representation is not required and a compromise with a finite number of neurons (*finite memory*) is acceptable, in general quite a few inputs are still required. This inevitably increases the complexity of the network, leads to an unnecessary large number of parameters, and hence slows down the estimation and training process [116].

This explains why the multilayer perceptron neural net can only be regarded as a nonlinear extension of autoregressive (AR) time series models Eqn. (4), but not a nonlinear extension of autoregressive moving-average (ARMA) models Eqn. (16).

$$\begin{aligned} x_t &= f(x_{t-1}, \ldots, x_{t-p}, \epsilon_{t-1}, ...\epsilon_{t-q}) + \epsilon_t \\ &= f(x_{t-1}, \ldots, x_{t-p}, x_{t-p-1}, ...) + \epsilon_t \end{aligned} \tag{16}$$

The finite memory problem of the multilayer perceptron neural net is well noticed by ANN researchers. In his celebrated article, Elman stated:

> "...the question of how to represent time in connection models is very important. One approach is to represent time *implicitly* by its effects on processing rather than *explicitly* (as in a spatial representation)". [76]: 179 (italics added)

The multilayer perceptron neural net tries to model time by giving it a spatial representation (that is, explicit) representation. What Elman suggests is to let time have an effect on the network response rather than represent it by an additional input dimension. Using an idea initiated by [95], Elman proposes an internal representation of memory by allowing the hidden unit patterns to be fed back to themselves. In this way, the network becomes *recurrent.*

The difference between the multilayer perceptron neural net (feed forward neural net) and the recurrent neural net can be shown as follows. For a multilayer perceptron neural network, Eqn. (4) can be re-formulated as Eqn. (6) (for a three-layer neural net – Fig. 1).

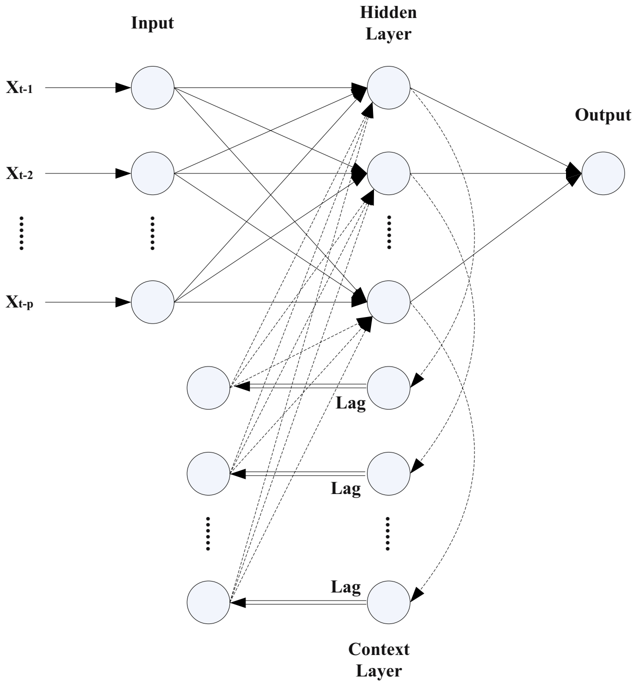

Fig. 2. The multilayer perceptron neural network model of a nonlinear AR process

A recurrent neural net – Eqn. (5) – can then be represented as:

$$x_t = h_2(w_0 + \sum_{j=1}^{l} w_j h_1(w_{0j} + \sum_{i=1}^{p} w_{ij} x_{t-i} + \sum_{m=1}^{l} \varpi_{mj} z_{m,t-1})) + \epsilon_t \qquad (17)$$

where

$$z_{m,t} = w_{0m} + \sum_{i=1}^{p} w_{im} x_{t-i} + \sum_{k=1}^{l} \varpi_{kj} z_{k,t-1}, \quad m = 1, \ldots, l \qquad (18)$$

Compared to the multilayer perceptron and radial basis function neural nets, the recurrent neural net has been much less explored in the economic and financial domains.[4] This is, indeed, a little surprising, considering the great exposure of its linear counterpart ARMA to economists.

2.4 Auto-Associative Neural Networks

While most economic and financial applications of neural networks consider the development of non-linear forecasting models, another important

[4] Some early applications can be found in [36] and [108].

consideration is dimensionality reduction and/or feature extraction. In this application, ANN can provide a nonlinear generalization of the conventional *principal component analysis* (PCA). The specific kind of ANN for this application is referred to as the *auto-associative neural network* (AANN).

The fundamental idea of principal component analysis is dimensionality reduction, which is a quite general problem when we are presented with a large number of correlated attributes, and hence a large number of redundancies. It is, therefore, a natural attempt to compress or store this original large data set into a more economical space by getting rid of these redundancies. Thus, on the one hand, we want to have a reduced space that is as small as possible; on the other hand, we still want to keep the original information. These two objectives are, however, in conflict when attributes with complicated relations are presented. Therefore, techniques to make the least compromise between the two become important.

To introduce AANN and its relationship to principal component analysis, let us consider the following two mappings,

$$\mathcal{G} : \mathbf{R}^m \to \mathbf{R}^f \tag{19}$$

and

$$\mathcal{H} : \mathbf{R}^f \to \mathbf{R}^m \tag{20}$$

where $\mathcal{G}$ and $\mathcal{H}$ are, in general, nonlinear vector functions with their components indicated as $\mathcal{G} = \{G_1, G_2, \ldots, G_f\}$ and $\mathcal{H} = \{H_1, H_2, \ldots, H_m\}$. To represent these functions with multilayer perceptron neural nets, let us rewrite Eqn. (6) as follows,

$$\begin{aligned} y_k &= G_k(x_1, \ldots, x_m) \\ &= h_2(w_{0k} + \sum_{j=1}^{l_1} w_{jk} h_1(w^e_{0j} + \sum_{i=1}^{m} w^e_{ij} x_i)), \quad k = 1, 2, \ldots, f \end{aligned} \tag{21}$$

and

$$\begin{aligned} \hat{x}_i &= H_i(y_1, \ldots, y_f) \\ &= h_4(w_{0i} + \sum_{j=1}^{l_2} w_{ji} h_3(w^d_{0j} + \sum_{k=1}^{f} w^d_{kj} y_k)), \quad i = 1, 2, \ldots, m \end{aligned} \tag{22}$$

All the notations used in Eqns. (21) and (22) share the same interpretation as those in Eqn. (6), except that superscripts e and d stand for the encoding and decoding maps, respectively. By combining the two mappings, we have a mapping from $\mathbf{X} = \{x_1, \ldots, x_m\}$ to its own reconstruction $\hat{\mathbf{X}} = \{\hat{x}_1, \ldots, \hat{x}_m\}$. Let X_n be the nth observation of X, and

$$\mathbf{X}_n = \{x_{n,1}, \ldots, x_{n,m}\} \tag{23}$$

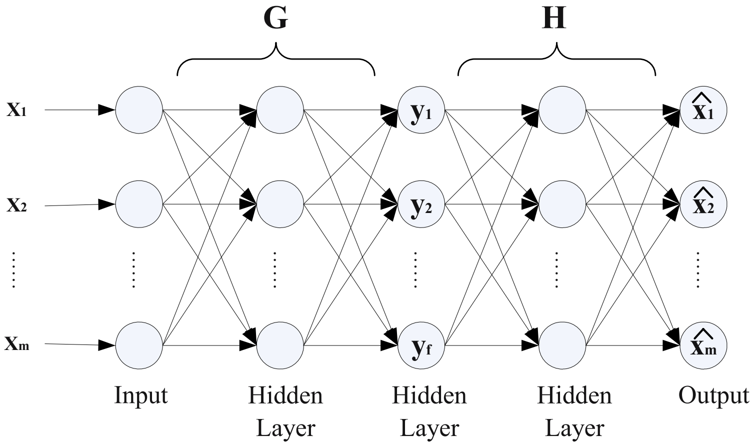

Fig. 3. The auto-associative neural network

Accordingly,

$$\hat{\mathbf{X}}_n = \{\hat{x}_{n,1}, \ldots, \hat{x}_{n,m}\} \tag{24}$$

Then minimizing the difference between observation $\mathbf{X}_n$ and its reconstruction $\hat{\mathbf{X}}_n$ over the entire set of N observations or

$$\min E = \sum_{n=1}^{N} \sum_{i=1}^{m} (x_{n,i} - \hat{x}_{n,i})^2 \tag{25}$$

by searching for the space of the connection weights and biases defines what is known as 'auto-association neural networks'. Briefly, auto-associative neural networks are feedforward nets, with *three hidden layers*, as shown in Fig. 3, trained to produce an approximation of the *identity mapping* between network inputs and outputs using backpropagation or similar learning procedures.

The third hidden layer – namely the output layer of the MLPN, (Eqn. (21)) – is also called the *bottleneck layer*. If the transfer functions h_i $(i = 1, 2, 3, 4)$ are all identical mappings, and we remove all the bias terms, then Eqn. (21) can be written as:

$$\begin{aligned} y_k &= G_k(x_1, \ldots, x_m) \\ &= \sum_{j=1}^{l_1} w_{jk} (\sum_{i=1}^{m} w_{ij}^e x_i) = \sum_{j=1}^{l_1} \sum_{i=1}^{m} w_{jk} w_{ij}^e x_i, \\ &= \sum_{i=1}^{m} \sum_{j=1}^{l_1} w_{jk} w_{ij}^e x_i, = \sum_{i=1}^{m} \beta_{i,k} x_i \quad k = 1, 2, \ldots, f, \end{aligned} \tag{26}$$

where

$$\beta_{i,k} = \sum_{j=1}^{l_1} w_{jk} w_{ij}^e \tag{27}$$

In matrix notation, Eqn. (26) can be written as:

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \begin{bmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1f} \\ \beta_{21} & \beta_{22} & \dots & \beta_{2f} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{m1} & \beta_{m2} & \dots & \beta_{mf} \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1f} \\ y_{21} & y_{22} & \dots & y_{2f} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \dots & y_{nf} \end{bmatrix}, \tag{28}$$

or simply

$$\mathbf{XB} = \mathbf{Y} \tag{29}$$

$\mathbf{X}$, $\mathbf{B}$ and $\mathbf{Y}$ correspond to the n-by-m, m-by-f, and n-by-f matrices in Eqn. (28), respectively. Likewise, Eqn. (22) can be simplified as:

$$\mathbf{YB}^* = \hat{\mathbf{X}} \tag{30}$$

$\mathbf{B}^*$ is the reconstruction mapping and is an f-by-m matrix, and $\hat{\mathbf{X}}$ is the reconstruction of $\mathbf{X}$, and hence is an n-by-m matrix.

Equation (29) and (30), together with the objective function (Eqn. (25)), define the familiar *linear* principal component analysis. To see this, we can decompose $\mathbf{X}$ as follows:

$$\mathbf{X} = \mathbf{YB}^* + \mathbf{E} = \mathbf{XBB}^* + \mathbf{E} = \mathbf{XP} + \mathbf{E} \tag{31}$$

where $\mathbf{P} = \mathbf{BB}^*$, and $\mathbf{E}$ is the reconstruction error. Then the PCA frequently presented to us takes the form of the following minimization problem.

$$\min_{\mathbf{P}} \| \mathbf{E} \| \tag{32}$$

It is known that the optimal solution of this problem (Eqn. (32)) has the rows of $\mathbf{P}$ being the eigenvectors corresponding to the f largest eigenvalues of the covariance matrix of $\mathbf{X}$. Therefore, we have shown how the self-associative neural network can be a nonlinear generalization of the familiar linear PCA, as well as how the linear PCA can be extended to the nonlinear PCA through a feedforward neural network with three hidden layers.

The concept of using a neural network with a bottleneck to concentrate information has been previously discussed in the context of *encoder/decoder* problems.[5] [119] indicates some directions for financial applications using nonlinear PCA.

[5] See [106] for a brief review.

2.5 Support Vector Machines

In the 1990s, based on results from *statistical learning theory* [149], an alternative to the artificial neural network was developed, in the form of the *support vector machine* (SVM). SVM was founded primarily by Vapnik, who contributed to the development of a general theory for minimizing the expected risk of losses using empirical data. Brief introductory material on the SVM can be found in [150], whereas [67] is a textbook devoted to the SVM.[6]

Support vector machines map non-linearly an n-dimensional input space into a high dimensional feature space.

$$\phi : V^n \rightarrow V^m \tag{33}$$

where V^n is an n-dimensional input vector space, and V^m is an m-dimensional feature vector space. Given a series of l historical observations:

$$(y_1, x_1), \ldots, (y_l, x_l) \tag{34}$$

where $y_i \in V^1$ and $x_i \in V^n$.

We approximate and estimate the functional relation between y_i and x_i by

$$y = f(x) =< w, \phi(x) > +b = \sum_{i=1}^{m} w_i \phi(x)_i + b \tag{35}$$

where $< . \, , \, . >$ denotes the inner product. The vector w and the constant b is to be determined by following the *structural risk minimization principle*, borrowed from statistical learning theory. It is interesting to note some similarities between the RBN and SVM, namely, Eqns. (9) and (35). However, there is also a noticeable difference. Consider an input x_i as a vector of three-dimensions: $(x_{i,1}, x_{i,2}, x_{i,3})$. Then for each neuron in the hidden layer of the RBN, they all share the same form as

$$(\varphi(x_{i,1}, x_{i,2}, x_{i,3}, c_1), \varphi(x_{i,1}, x_{i,2}, x_{i,3}, c_2), ...) \tag{36}$$

while being associated with different centers. However, each neuron in the hidden layer of the SVM may actually take different inputs. For example, the first neuron takes the first two inputs, but the second takes the last two as

$$(\phi_1(x_{i,1}, x_{i,2}), \phi_2(x_{i,2}, x_{i,3}), ...) \tag{37}$$

Also, notice that the transfer functions, $\varphi(\ \ )$ are the same for each neuron in the RBN, but in general are different for the SVM as $\phi_1, \phi_2, ...$

[6] Financial applications have kept on expanding; the interested reader can find some useful references directly from the SVM website: http://www.svms.org/

In the case where the y_i are categorical, such as $y_i \in \{-1, 1\}$, the minimization process also determines a subset of $\{x_i\}_{i=1}^{l}$, called *support vectors*, and the SVM when constructed has the following form.

$$f(x) = \sum_{s} y_i \alpha_i^* < \phi(x_s), \phi(x) > +b^* \tag{38}$$

where α_i^* and b^* are the coefficients satisfying the structural risk minimization principle, and s is the set of all support vectors.

The category assigned to the observation x, 1 or -1, will then be determined by the sign of $f(x)$.

$$y = \begin{cases} 1, & if \;\; f(x) > 0 \\ -1, & if \;\; f(x) < 0 \end{cases} \tag{39}$$

Eqns. (38) and (39) are the SVM for the classification problem. A central concept of the SVM is that one does not need to consider the feature space in explicit form; instead, based on the Hilbert-Schmidt theory, one can use the *kernel function*, $K(x_s, x)$, where

$$K(x_s, x) =< \phi(x_s), \phi(x) > \tag{40}$$

Therefore, the SVM is also called the *kernel machine*. Eqn. (38) can then be rewritten as

$$f(x) = \sum_{s} y_i \alpha_i^* K(x_s, x) + b^* \tag{41}$$

Following a similar procedure, one can construct an SVM for regression problems as follow:

$$f(x) = \sum_{i=1}^{l} (\alpha_i^* - \beta_i^*) K(x, x_i) + b^* \tag{42}$$

where α_i^*, β_i^* and b^* are the coefficients minimizing the corresponding objective functions.

In addition to the functional form $f(\mathbf{x})$, the second important issue is the set of variables $\mathbf{x}$ itself, and one has to deal naturally with the problem known as *variable selection* or *feature selection*. The involvement of irrelevant variables or features may lead to poor generalization capability.

2.6 Self-Organizing Maps and k-means

In the social and behavioral sciences, the ability to recognize patterns is an essential aspect of human heuristic intelligence. Herbert Simon, a Nobel Prize Laureate in Economics (1978), considered pattern recognition to be critical and advocated the need to pay much more explicit attention to the teaching of pattern recognition principles. In the financial market, chartists appear to have been good at performing pattern recognition for many decades, yet little academic research has been devoted to a systematic study of these kinds of activities. On the contrary, sometimes it has been treated as nothing more than astrology, and hardly to be regarded as a rigorous science.

The Self-Organizing Map was invented by Kohonen [101]. It has been applied with great success to many different engineering problems and to many other technical fields. [71] was the first volume to demonstrate the use of the SOM in finance.

Self-organizing maps (SOMs) solve the pattern recognition problem which deals with a class of unsupervised neural networks. Basically, the SOM itself is a two-layer neural network. The input layer is composed of p cells, one for each system input variable. The output layer is composed of neurons which are placed on n-dimensional lattices (the value of n is usually 1 or 2). The SOM adopts so-called *competitive learning* among all neurons. Through competitive learning, the neurons are tuned to represent a group of input vectors in an organized manner.

k-means clustering, developed by [115], is a widely used *non-hierarchical clustering algorithm* that groups data with similar characteristics or features together. k-means and SOMs resemble each other. They both involve minimizing some measure of dissimilarity, called the cost function, in the samples within each cluster. The difference between the k-means and the SOM lies in their associated cost function to which we now turn. Consider a series of n observations, each of which has m numeric attributes:

$$\mathbf{X}_1^m, \mathbf{X}_2^m, \ldots, \mathbf{X}_n^m, \quad \mathbf{X}_i^m \in \mathbf{R}^m \;\; \forall\, i = 1, 2, \ldots, n \tag{43}$$

where

$$\mathbf{X}_i^m \equiv \{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\}. \;\; x_{i,l} \in \mathbf{R}, \forall\, l = 1, 2, \ldots, m \tag{44}$$

The k-means clustering is to find a series of k clusters, the centroids of which are denoted, respectively, by

$$\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_k, \quad \mathbf{M}_j \in \mathbf{R}^m, \quad \forall j = 1, 2, \ldots, k \tag{45}$$

such that each of the observations is assigned to one and only one of the clusters with minimal cost, and with cost function being defined as follows:

$$C_{k-means} = \sum_{i=1}^{n} \sum_{j=1}^{k} d(\mathbf{X}_i^m, \mathbf{M}_j) \cdot \delta_{i,j} \tag{46}$$

where $d(\mathbf{X}_i^m, \mathbf{M}_j)$ is the standard Euclidean distance between $\mathbf{X}_i^m$ and $\mathbf{M}_j$,[7] and $\delta_{i,j}$ is the delta function:

[7] Standard Euclidean distance assumes that the attributes are normalized and are of equal importance. However, this assumption may not hold in many application domains. In fact, one of the main problems in learning is to determine *which* are the important features.

$$\delta_{i,j} = \begin{cases} 1, \ if \quad \mathbf{X}_i^m \in \mathbf{Cluster}_j \\ 0, \ if \quad \mathbf{X}_i^m \notin \mathbf{Cluster}_j \end{cases} \tag{47}$$

To minimize the cost function (Eqn. (46)), one can begin by initializing a set of k cluster centroids. The positions of these centroids are then adjusted iteratively by first assigning the data samples to the nearest clusters and then re-computing the centroids.

Corresponding to Eqn. (46), the cost function associated with SOM can be roughly treated as follows[8]

$$C_{SOM} = \sum_{i=1}^{n} \sum_{j=1}^{k} d(\mathbf{X}_i^m, \mathbf{M}_j) \cdot h_{w(\mathbf{X}_i^m),j} \tag{48}$$

where $h_{w(\mathbf{X}_i^m),j}$ is the neighborhood function or neighborhood kernel, and $w_{\mathbf{X}_i^m}$ – the winner function – outputs the cluster whose centroid is nearest to the input $\mathbf{X}_i^m$.

In practice, the neighborhood kernel is chosen to be wide at the beginning of the learning process to guarantee global ordering of the map, and both its width and height decrease slowly during learning. For example, the Gaussian kernel whose variance monotonically decreases with iteration times t is frequently used.[9] By comparing Eqn. (46) with Eqn. (48), one can see in SOM the distance of each input from all of the centroids weighted by the neighborhood kernel h, instead of just the closest one being taken into account.

Despite its greater simplicity, the economic and financial applications of k-means are surprisingly much less available than those of SOM and KNN. k-means have occasionally been applied to classify hedge funds [68], listed companies [128], and houses [93], but it can also be applied to the classification of trajectories of financial time series. To see this, we rewrite Eqns. (43) and (44) to fit the notations used in the context of time series:

$$\mathbf{X}_1^m, \mathbf{X}_2^m, \ldots, \mathbf{X}_T^m, \quad \mathbf{X}_t^m \in \mathbf{R}^m, \quad \forall\, t = 1, 2, \ldots, T \tag{49}$$

$$\mathbf{X}_t^m \equiv \{x_t, x_{t-1}, \ldots, x_{t-m}\}, \quad x_{t-l} \in \mathbf{R}, \forall\, l = 0, 1, \ldots, m-1 \tag{50}$$

$\mathbf{X}_t^m$ is a windowed series with an immediate past of m observations, also called the m-history. Eqn. (49), therefore, represents a sequence of T m-histories which are derived from the original time series, $\{x_t\}_{t=-m+1}^{T}$, by moving the m-long window consecutively, one step at a time. Accordingly, the end-product of applying k-means or SOMs to these windowed series is a number of centroids $\mathbf{M}_j$, which represents a specific shape of an m-long trajectory, also known as 'charts' by technical analysts.[10]

[8] The rigorous mathematical treatment of the SOM algorithm is extremely difficult in general – see [102].

[9] For details, see [51] Chap. 8: 205.

[10] For example, see the charts presented in [44]: 206–207.

Then the essential question pursued here is whether we can meaningfully cluster the windowed financial time series $\mathbf{X}_t^m$ by the k associated geometrical trajectories, $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_k$. The clustering work can be meaningful if it can help us predict the future. In other words, conditional on a specific trajectory, we can predict the future better than without being provided this information, for example,

$$Prob(|\xi_{t+1}| > |\epsilon_{t+1}|) > 0.5 \tag{51}$$

where

$$\xi_{t+1} = x_{t+1} - E(x_{t+1}) \tag{52}$$

and

$$\epsilon_{t+1} = x_{t+1} - E(x_{t+1} | \mathbf{X_t^m} \in \mathbf{Cluster_j}), \quad t > T \tag{53}$$

The conditional expectations above are made with the information of the trajectory (the cluster).

2.7 K Nearest Neighbors

In 1998, a time-series prediction competition was held during the *Intl. Workshop on Advanced Black-Box Techniques for Nonlinear Modeling*. The data to be predicted were available from November 1997 through April 1998 at Leuven. The data was generated from a generalized Chua's circuit, a well-known chaotic dynamic system. Seventeen entries had been submitted before the deadline. The winner of the competition turned out to be James McNames, and the strategy he used was the *nearest trajectory algorithm*. By using this algorithm to fast nearest neighbor algorithms, McNames was able to make an accurate prediction up to 300 points in the future of the chaotic time-series. At first sight, this result may be a surprise for some, because KNN is not technically demanding in contrast to many other well known tools as introduced in this Chapter, nevertheless it could outperform many other familiar advanced techniques, such as neural nets, wavelets, Kohonen maps (SOM), and Kalman filters in that competition.[11]

KNN can be related to *decision trees*. What makes them different is that the latter have categories $A_1, \ldots, A_n$ to host input variables $\mathbf{X_t^m}$, while the former have $\mathbf{X_t^m}$ itself as a center of a hosting category, which will invite its own *neighbors*, $\mathbf{X_s^m}$ ($s < t$), by ranking the *distance* $||\mathbf{X_t^m} - \mathbf{X_s^m}||$ over all $s < t$ from the closest to the farthest. Then the k closest $\mathbf{X_s^m}$s will constitute the neighbors of $\mathbf{X_t^m}$, $\mathcal{N}(\mathbf{X_t^m})$. Now, for the purpose of predicting x_{t+1}, one can first study the functional relation between x_{s+1} and $\mathbf{X_s^m}$, $\forall s \in \mathcal{N}(\mathbf{X_t^m})$, in other words,

$$x_{s+1} = f_t(\mathbf{X_s^m}), s \in \mathcal{N}(\mathbf{X_t^m}) \tag{54}$$

[11] For details of the competition report, see [140].

One then forecasts x_{t+1} based on $\hat{f}_t$, an estimation of f_t,

$$\hat{x}_{t+1} = \hat{f}_t(\mathbf{X_t^m}) \tag{55}$$

Let's make a brief remark on what makes KNN different from conventional time-series modeling techniques. Conventional time-series modeling, known as the Box-Jenkins approach, is a *global* model, which is concerned with the estimation of the function, be it linear or non-linear, in the following form:

$$x_{t+1} = f(x_t, x_{t-1}, \ldots, x_{t-m}) + \epsilon_t = f(\mathbf{X_t^m}) + \epsilon_t \tag{56}$$

by using all of the information up to t – that is, $\mathbf{X_s^m} \forall s \leq t$ – and the estimated function $\hat{f}$ is assumed to hold for every single point in time. As a result, what will affect x_{t+1} most is its immediate past $x_t, x_{t-1}, \ldots$ under the law of motion estimated by all available samples.

For KNN, while what affects x_{t+1} most is also its immediate past, the law of motion is estimated *only* with *similar* samples, *not all* samples. The estimated function $\hat{f}_t$ is hence assumed to only hold for that specific point in time. Both KNN and SOM challenge the conventional Box-Jenkins methodology by characterizing the hidden patterns in a different form. In their formulation, hidden patterns are not characterized by time location, but by topological trajectories.

Technical issues involved here are the choice of distance function $d(\mathbf{X_t^m}, \mathbf{X_s^m})$, choice of functional form f_t, choice of the number of neighbors k, and choice of the embedding dimension m.

2.8 Instance-Based Learning

KNN can be regarded as a special case of a broader class of algorithms, known as *instance-based learning* (IBL). To see this, let us use the notations introduced in Sect. 2.6, and use the time series prediction problem as an illustration.

Consider Eqn. (53). We have been given information regarding a time series up to time t, and we wish to forecast the next by using the current m-history, $\mathbf{X}_t^m$. In SOM or KNN, we will first decide which cluster $\mathbf{X}_t^m$ belongs by checking $d(\mathbf{X}_t^m, \mathbf{M}_j)$ for all j ($j = 1, 2, \ldots, k$), and use the forecast model associated with that cluster to forecast x_{t+1}. In other words, forecasting models are tailored to each cluster, say, $\hat{f}_j$ ($j = 1, 2, \ldots, k$).[12] Then

$$\hat{x}_{t+1} = \hat{f}_{j^*}(\mathbf{X}_t^m), \quad if \quad j^* = \arg\min_j d(\mathbf{X}_t^m, \mathbf{M}_j) \quad j = 1, 2, \ldots, k \tag{57}$$

[12] The notation $\hat{f}$ is used, instead of f, to reserve f for the true relation, if it exists, and in that case, $\hat{f}$ is the estimation of f. In addition, there are variations when constructing Eqn. (57) – see [44].

KNN, however, does not have such established clusters $\mathbf{M}_j$. Instead, it forms a cluster based on each $\mathbf{X}_t^m$, $\mathcal{N}(\mathbf{X}_t^m)$, as follows.

$$\mathcal{N}(\mathbf{X}_t^m) = \{s \mid Rank(d(\mathbf{X}_t^m, \mathbf{X}_s^m)) \leq k, \forall s < t\} \tag{58}$$

In other words, $\mathbf{X}_t^m$ itself serves as the centroid of a cluster, called the *neighborhood* of $\mathbf{X}_t^m$, $\mathcal{N}(\mathbf{X}_t^m)$. It then invites its *k nearest neighbors* to be the members of $\mathcal{N}(\mathbf{X}_t^m)$ by ranking the distance $d(\mathbf{X}_t^m, \mathbf{X}_s^m)$ over the entire community

$$\{\mathbf{X}_s^m \mid s < t\} \tag{59}$$

from the closest to the farthest.

Then, by assuming a functional relation, f, between x_{s+1} and $\mathbf{X}_s^m$ and using only the observations associated with $\mathcal{N}(\mathbf{X}_t^m)$ to estimate this function f_t,[13] one can construct the tailor-made forecast for each x_t,

$$\hat{x}_{t+1} = \hat{f}_t(\mathbf{X}_t^m) \tag{60}$$

In practice, the function f used in Eqn. (60) can be very simple, either taking the *unconditional mean* or the *conditional mean*. In the case of the latter, the mean is usually assumed to be linear. In the case of the unconditional mean, one can simply use the simple average in the forecast,

$$\hat{x}_{t+1} = \frac{\sum_{s \in \mathcal{N}(\mathbf{X}_t^m)} x_{s+1}}{k} \tag{61}$$

but one can also take the weighted average based on the distance of each member.

The same idea can be applied to deal with the linear conditional mean (linear regression model): we can either take the ordinal least squares or the weighted least squares.[14]

From the above description, we find that KNN is different from k-means and SOM in the sense that, not just the forecasting function, but also the cluster for KNN is tailor-made. This style of tailor-made learning is known as *lazy learning* in the literature [2]. It is called 'lazy' because learning takes place when the time comes to classify a new instance, say $\mathbf{X}_{T+t}^m$, rather than when the *training set*, Eqn. (49), *is processed*, say T.[15]

[13] Even though the functional form is the same, the coefficients can vary depending on $\mathbf{X}_t^m$ and its resultant $\mathcal{N}(\mathbf{X}_t^m)$. Accordingly, we add a subscript t as f_t to make this time-variant property clear.

[14] Details can be found in [78].

[15] Note that a fixed T in Eqn. (49) implies a fixed training set without increments. A non-incremental training set can be typical for using k-means or SOM. However, KNN learning, also known as *rote learning*, memorizes everything that happens up to the present; therefore, the 'training set' (memory) for KNN grows with time.

To make this clear, consider two types of agents: the k-means agent and the KNN agent. The k-means agent learns from the history before new instances come, and the resultant knowledge from learning is represented by a set of clusters, which is *extracted* from a set of historical instances. Based on these clusters, some *generalization pictures* are already produced before the advent of new instances, say $\mathbf{X}^m_{T+t}$. The KNN agent, however, is not eager to learn. While he does store every instance observed, he never tries to extract knowledge (general rules) from them. In other words, he has the simplest form of 'learning', that is, rote learning (plain memorization). When the time $T+t$ comes and a new instance $\mathbf{X}^m_{T+t}$ is encountered, his memory is then searched for the historical instances that most strongly resemble $\mathbf{X}^m_{T+t}$.

As stated previously, KNN, as a style of rote learning, stores all the historical instances, as shown in Eqn. (59). Therefore, amounts of storage increase with time. This may make the nearest-neighbor calculation unbearably slow. In addition, some instances may be regarded as redundant with regard to the information gained. This can be particularly the case when KNN is applied to *classification* rather than regression or time series forecasting. For example, if we are interested not in x_{t+1} itself, but in whether x_{t+1} will be greater than x_t – namely, whether x_t will go up or go down, then some regions of the instance space may be very stable with regard to class – for instance, up (1) or down (0) – and just a few exemplars are needed inside stable regions. In other words, we do not have to keep all historical instances or training instances. The *storage-reduction algorithm* is then used to decide which instances in Eqn. (59) to save and which to discard. This KNN with the storage-reduction algorithm is called *instance-based learning* (IBL) and was initiated by [3].[16]

The addition of a storage-reduction algorithm to KNN is also interesting from the perspectives of both neural sciences and economics. Considering the brain with its limited capacity for memory, then an essential question to ask is how the brain deals with increasing information by not memorizing all of it or by forgetting some of it. How does it perform pruning? This is still a non-trivial issue pursued by neural scientists today. The same issue can interest economists as well, because it concerns the efficient use of limited space. A recent study on reward-motivated memory formation by neural scientists may provide an economic foundation for the memory formation [1].[17]

In this vein, the *marginal productivity* of the new instance in IBL can be considered as the reward. The marginal productivity of an instance can

[16] As a matter of fact, the storage-reduction algorithms are not just to deal with the *redundancy* issue, but also the *noise-tolerance* issue. [3] distinguish the two by calling the former *memory updating functions*, and the latter *noise-tolerant algorithms*.

[17] [1] report brain-scanning studies in humans that reveal how specific *reward-related brain regions* trigger the brain's learning and memory regions to promote memory formation.

be defined by its contribution to enhance the capability to perform a correct classification. For those instances which have low marginal productivity, it will be discarded (not remembered), and for those already stored instances, if their classification performances are poor, they will be discarded, too (forgotten). In this way, one can interpret the mechanism of the pruning algorithms or the storage-reduction algorithms used in computational intelligence in the fashion of neural economics.

3 Evolutionary Computation

The second important pillar of computational intelligence is so called *evolutionary computation* (EC). EC uses *Nature* as an inspiration. While it also has a long history of utilization in economics and finance, it is, relatively speaking, the 'new kid on the block', as compared with neural networks, and even more so with fuzzy logic. It has also drawn less attention from economists and financial analysts than the other two approaches. By comparison, there are already about a dozen books or volumes on the economic and financial applications using fuzzy logic and neural nets. In the area of EC, there are only three volumes edited for economists and financiers [25,39,40]. Evolutionary computation is generally considered to be a consortium of *genetic algorithms* (GA), *genetic programming* (GP), *evolutionary programming* (EP) and *evolutionary strategies* (ES).

The history of evolutionary computation can be traced back to the mid-1960s, where evolutionary strategies were originated by Rechenberg [129], Schwefel [134] and Bienert at the Technical University of Berlin. The development of genetic algorithms started with Holland at the University of Michigan, and evolutionary programming was originated by Fogel [80] at the University of California at Los Angeles.[18] Despite their non-trivial differences, they share the common structure shown in Fig. 4.

Evolutionary computation starts with an initialization of a population of individuals (solution candidates), called $P(0)$, with a *population size* to be supplied by the users. These solutions will then be evaluated based on an *objective function* or a *fitness function* determined by the problem of interest. Continuation of the procedure will hinge on the *termination criteria* supplied by the users. If these criteria are not met, then we move to the next stage or *generation* by adding 1 to the time counter ($t \rightarrow t+1$). Two major operators are conducted to form the new generation, which can be regarded as a *correspondence*, as follows,

$$F_{s_2} \circ F_a \circ F_{s_1}(P(t)) = P((t+1)) \tag{62}$$

where F_{s_1} and F_{s_2} denotes *selection*, and F_a denotes *alteration*.

[18] For a description of the birth of EC, see [75], [79], and [135].

```
begin
    t := 0;
    Initialize P(t);
    evaluate P(t);
    while not terminating do
    begin
        M(t) := select-mates(P(t));
        O(t) := alternation(M(t));
        evaluate(O(t));
        P(t+1) := select(O(t)∪P(t));
        t := t+1;
    end
end
```

Fig. 4. Evolutionary computation (EC) pseudo-algorithm

The main purpose of the first-stage selection, F_{s_1}, is to form a mating pool (a collection of parents), $M(t)$, which can in turn be used to breed the new generation:

$$F_{s_1}(P(t)) = M(t). \tag{63}$$

Once the mating pool is formed, F_a is applied to generate *offspring*, $O(t)$, from these parents. Two major steps (*genetic operators*) are involved here, namely, *recombination* (*crossover*), denoted by F_r, and *mutation*, denoted by F_m, which shall be described in detail later.

$$F_a(M(t)) = F_m \circ F_r(M(t)) = O(t). \tag{64}$$

These offspring will be first evaluated, then enter the second-stage selection *with* or *without* their parents $P(t)$. Finally, the new generation $P(t+1)$ is formed as a result of the second-stage selection.

$$F_{s_2}(O(t) \cup P(t)) = P((t+1)). \tag{65}$$

After that, we go back to the beginning of the loop, and then check the termination criteria to see whether to stop or to start another generation of runs – see Fig. 5 for the evolution loop.

Based on the description above, it is perhaps beneficial to have the seven major components of evolutionary algorithms listed as follows for quick reference:

1. individuals and their representations,
2. initialization,
3. fitness evaluation,
4. selection,

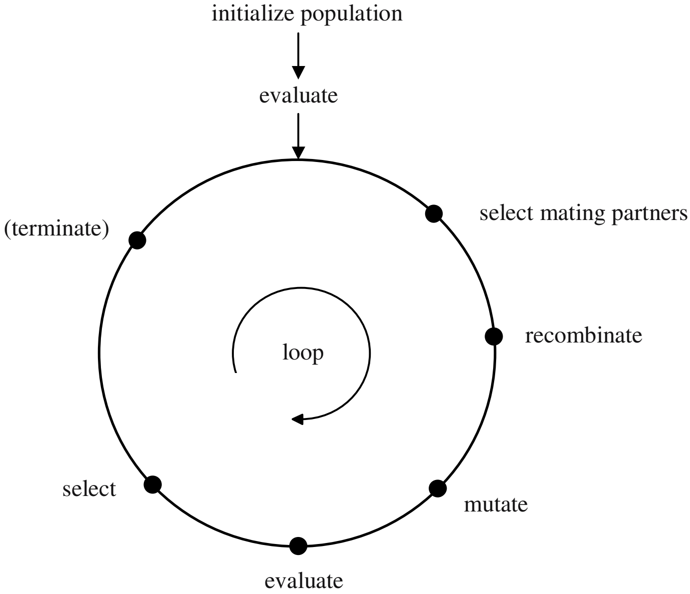

Fig. 5. The evolutionary loop

5. mutation,
6. recombination,
7. replacement.

3.1 Evolutionary Strategies

We shall illustrate each of these components mainly within the context of *evolutionary strategies*. Individuals are also called *chromosomes*. The individual in ES is represented as a pair of real-valued vectors $v = (x, \sigma)$, where the x represent a point in the solution space, and σ is a standard deviation vector that determines the mutation step size. Generally, σ is also called the *strategy parameter* in ES, and x is called the *object variable*.

The ES population size of is usually characterized by two parameters μ and λ. The former is the population size of $P(t)$, whereas the later is the population size of $O(t)$. Selection of F_{s_1} is much more straightforward in ES than in GA. Usually, it takes the whole $P(t)$ as the mating pool and parents are randomly selected therein. However, selection of F_{s_2} in ES can be more intriguing. There are two F_{s_2} schemes in ES, known as the $(\mu + \lambda)$ (Plus) scheme and the (μ, λ) (Comma) scheme. In the $(\mu + \lambda)$ scheme, μ individuals produce λ offspring, and a new population is formed by selecting μ individuals from $\mu + \lambda$. In the (μ, λ) scheme, μ individuals produce λ offspring, and a new population is formed by selecting μ individuals from the λ offspring. There is generally no constraint for μ and λ for the $(\mu + \lambda)$ scheme, but for the

(μ, λ) scheme, to make selection meaningful, μ has to be strictly less than λ; moreover, $\lambda/\mu \approx 7$ is an ideal ratio.

Mutation is considered the major ES operator for altering chromosomes. Mutation is applied to this individual to perturb real-valued parameters. If we let v be the parent randomly selected from $P(t)$, then mutation on v can be described as follows:

$$v' = (x', \sigma') = (f_{m_x}(x), f_{m_\sigma}(\sigma)) \tag{66}$$

where

$$f_{m_x}(x) = x + N(0, (\sigma')^2) \tag{67}$$

and

$$f_{m_\sigma}(\sigma) = \sigma \exp(\tau N(0,1)) \tag{68}$$

$N(0, \sigma^2)$ denotes the normal distribution with mean 0 and variance σ^2.[19] Notice that in implementation, Eqn. (68) has to be computed before Eqn. (67). This is because x' is obtained by mutating x with the new standard deviation σ'.[20]

Recombination operators compose new chromosomes from corresponding parts of two or more chromosomes. For the binary case, two chromosomes $v_1 = (x_1, \sigma_1^2)$ and $v_2 = (x_2, \sigma_2^2)$ are to be recombined by an operator f_r. We can describe the composition of a new chromosome v' as follows:

$$v' = (x', \sigma') = (f_{r_x}(x_1, x_2), f_{r_\sigma}(\sigma_1^2, \sigma_2^2,)) \tag{69}$$

Each element of the object and strategy parameter is a recombination of the respective entries v_1 and v_2. There is great variation of f_{r_x} and f_{r_σ}. In the ES literature, they are differentiated by the terms *discrete* or *intermediate*, *dual* (sexual) or *global* (panmictic). With a *discrete* recombination function, one of the corresponding components is chosen at random and declared the new entry. With an intermediate recombination, a linear combination of the corresponding components is declared the new entry. More formally, consider x' as an example:

$$x' = \begin{cases} x_1 \quad or \quad x_2 & discrete \\ \chi x_1 + (1-\chi) x_2 & intermediate \end{cases} \tag{70}$$

where $\chi \in [0, 1]$ denotes a uniform random variable.

[19] Here, for simplicity, we assume that x is a real-valued number. In a more general setting, the variable x can be a vector; in that case, σ should be replaced by the variance-covariance matrix Σ.

[20] In Eqn. (67), $(\sigma')^2$ is determined randomly. There is, however, some way to make it adaptive. For example, in the $(1+1)$-ES case, one has the famous 1/5-success rule. $(\sigma')^2$ can also be determined in a self-adaptive way. In that case, the learning rate τ can be set as a function of time. For details, see [135].

So far we have only considered a one-dimensional x. An n-dimensional x can further complicate the recombination function, and that is where the terms *dual* and *global* come from. *Dual* means that two parents are chosen at random for the creation of the offspring. *Global* means that one parent is chosen anew for *each component* of the offspring.

$$x'_i = \begin{cases} x_{1,i} \;\; or \;\; x_{2,i} & discrete, dual \\ x_{1,i} \;\; or \;\; x_{(2),i} & discrete, global \\ \chi x_{1,i} + (1-\chi) x_{2,i} & intermediate, dual \\ \chi x_{1,i} + (1-\chi) x_{(2),i} & intermediate, global \end{cases} \tag{71}$$

where $x_{(2),i}$ indicates that parent 2 is chosen anew for each vector component i, $(i = 1, 2, \ldots, n)$.

3.2 Evolutionary Programming

While evolutionary programming (EP) was proposed about the same time as evolutionary algorithms, their initial motives were quite different. Evolutionary strategies were developed as a method to solve *parametric optimization problems*, whereas evolutionary programming was developed as a method to simulate *intelligent behavior*. Lacking a capability to predict, an agent cannot adapt its behavior to meet the desired goals, and success in predicting an environment is a prerequisite for intelligent behavior. As Fogel puts it:

> "Intelligent behavior is a composite ability to predict one's environment coupled with a translation of each prediction into a suitable response in the light of some objective". ([82]: 11)

During the early stage, the prediction experiment can be illustrated with a sequence of symbols taken from a finite alphabet, say, a repeating sequence '(101110011101)*' from the alphabet $\{0, 1\}$. The task then is to create an algorithm that would operate on the observed indexed set of symbols and produce an output symbol that agrees with the next symbol to emerge from the environment. Fogel took *finite state automata* (FSA) as the machine to predict the sequence. A FSA is a device which begins in one state and upon receiving an input symbol, changes to another state according to its current state and the input symbol. EP was first proposed to evolve a population of finite state machines that provides successively better predictions.

3.3 Genetic Programming and Genetic Algorithms

While genetic programming has been applied to economic modeling for more than half a decade, its relevance to the nature of economics has not been fully acknowledged. In the most sympathetic situations, it is regarded as nothing but *alchemy*. In unsympathetic situations, it is notorious for its *black-box* operation. Sometimes, the process and results are so complicated that economists can hardly consider it relevant and interesting. This Section is intended to deliver a simple but strong message: *genetic programming is not just another fancy technique exploited by the unorthodox, but could be a faithful language to express the essence of economics.* In particular, it provides evolutionary economists with a way to *substantiate* some features which distinguish them from mainstream economists.

An Evolving Population of Decision Rules

Let's start from the most fundamental issue: *why is genetic programming relevant*? Lucas provided a notion of an *economic agent*.

> "In general terms, we view or model an individual as *a collection of decision rules* (rules that dictate the action to be taken in given situations) and *a set of preferences* used to evaluate the outcomes arising from particular situation-action combinations". [114]: 217 (italics added)

Immediately after this *static description* of an economic agent, Lucas described an *adaptive* (*evolutionary*) version:

> "These decision rules are continuously under review and revision: new decision rules are tried and tested against experience, and rules that produce desirable outcomes supplant those that do not". (*Ibid*: 217).

So, according to Lucas, the essence of an economic agent is *a collection of decision rules which are adapting (evolving) based on a set of preferences.* In short, it is the idea of an 'evolving population'.

Suppose that an evolving population is the essence of the economic agent, then it seems important to know whether we economists know any operational procedure to substantiate this essence. Back in 1986, the answer was absolutely 'no'. That certainly does not mean that we did not know anything about evolving *decision rules*. On the contrary, since the late 1970s, the literature known as 'bounded rationality in macroeconomics' has introduced a number of techniques to evolve a single decision rule (a single equation or a single system of equations): recursive regression, Kalman filtering, and Bayesian updating, to name a few; [132] made an extensive survey of this subject. However, these techniques shed little light on how to build a Lucasian agent, especially since what we wanted to evolve was not a single decision rule but a population of decision rules.

In fact, it may sound a little surprising that economists in those days rarely considered an individual as a population of decision rules, not to mention attending to the details of its evolution. Therefore, all the basic issues pertaining to models of the evolving population received little, if any, attention.

For example, how does the agent *initialize* a population of decision rules? Once the agent has a population of decision rules, which one should they follow? Furthermore, in what ways should this population of decision rules 'be continuously under review and revision'? Should we review and revise them one by one because they are independent, or modify them together because they may correlate with each other? Moreover, if there are some 'new decision rules to be tried', how do we generate (or find) these new rules? What are the relations between these new rules and the old ones? Finally, it is also not clear how 'rules that produce desirable outcomes should supplant those that do not.'

There is one way to explain why economists are not interested in, and hence not good at, dealing with a population of decision rules: economists used to derive the decision rule for the agent *deductively*, and the deductive approach usually leads to only one solution (decision rule), which is the *optimal* one. There was simply no need for a population of decision rules.

Genetic Algorithms and Classifier Systems

We do not know exactly when or how the idea of the *evolving population of decision rules* began to attract economists, but Holland's contribution to *genetic algorithms* definitely exerted a great influence. Genetic algorithms simulate the biological evolution of a society of computer programs, each of which is represented by a chromosome or, normally, a string of binary ones and zeros. Each of these computer programs can be matched to a solution to a problem. This structure provides us with an operational procedure of the Lucasian agent. First, a collection of decision rules are now represented by a society of computer programs (a society of strings of binary ones and zeros). Second, the review and revision process is implemented as a process of natural selection.

While genetic algorithms have had a great impact on computer science, mathematics, and engineering since the early 1980s, their implications for social sciences were not acknowledged until the late 1980s. In 1987, Axelrod, a political scientist at the University of Michigan, published the first application of the GA to the social sciences [21]. A year later, the first PhD dissertation that applied GAs to the social sciences was completed by John Miller from, not surprisingly, the University of Michigan. The issue addressed by Axelrod and Miller is the well-known *repeated prisoner's dilemma*. In addition to these two early publications, perhaps the most notable event that brought GAs into economics was the invited speech by John Holland at an economic conference at the Santa Fe Institute in the autumn of 1987. Among the audience were some of the most prestigious contemporary economists, including Kenneth Arrow, Thomas Sargent, Hollis Chenery, Jose Scheinkman, and Brian Arthur. In his lecture entitled 'The global economy as an adaptive process', Holland introduced to the economics circle the essence of genetic algorithms as 'building blocks'.

A building block refers to the specific pattern of a chromosome – that is, an essential characteristic of a decision rule. There is a formal word for this in the genetic algorithm; it is called a *schema*. In the genetic algorithm, a schema is regarded as the basic unit of learning, evolution, and adaptation. Each decision rule can be defined as a combination of some schemata. The review and revision process of decision rules is nothing more than a search for the right combination of those, possibly infinite, schemata. To rephrase Lucas's description in Holland's words, "economic agents are constantly revising and rearranging their building blocks as they gain experience". Not only do genetic algorithms make the Lucasian economic agent implementable, but they also enrich its details.

After a gradual spread and accumulation of knowledge about GA among economists, modeling economic agents with an evolving population of decision rules finally began to increase in the 1990s. To the best of this author's knowledge, the first refereed journal article was [117]. This paper is follow-up research to that of [100]. In a simple barter economy, Kiyotaki and Wright found that low storage costs are not the *only* reason why individuals use money. The other one is that money makes it easier to find a suitable partner. Replacing the rational agents in the Kiyotaki-Wright environment with *artificially intelligent* agents, [117], however, found that goods with low storage costs play the dominating role as a medium of exchange.

The population of decision rules used to model each agent is a *classifier system*, another contribution made by Holland in the late 1970s. A classifier system is similar to the Newell-Simon type expert system, which is a population of *if..then* or *condition-action* rules. However, the classical expert system is not adaptive. What Holland did with the classifier system was to apply the idea of competition in the market economy to a society of if..then rules. Market-like competition is implemented by way of a formal algorithm known as the *bucket-brigade algorithm,* credit rules generating good outcomes and debit rules generating bad outcomes. This accounting system is further used to resolve conflicts among rules. The shortcoming of the classifier system is that it cannot automatically generate or delete rules. Nonetheless, by adding a genetic algorithm on top of the bucket brigade and rule-based system, one can come up with something similar to a Lucasian agent, which not only learns from experience, but can be *spontaneous* and *creative*.

While Holland's version of the adaptive agent is much richer and more implementable than the Lucasian economic agent, and the work was already completed before the publication of [91], its formal introduction to economists came five years after the publication of [114]. In 1991, Holland and Miller published a sketch of the *artificial adaptive agent* in the highly influential journal *American Economic Review*. The first technique to implement the

Lucasian economic agent was finally ‘registered’ in economics, and genetic algorithms and classifier systems were formally added to economic analysts’ toolkits. Is five years too long? *Maybe not,* given that ‘Economic analysis has largely avoided questions about the way in which economic agents make choices when confronted by a perpetually novel and evolving world’ ([92]: 365).

What’s next? If the Lucasian economic agent is a desirable incarnation of the economic agent in economic theory, and if Holland’s artificial adaptive agent is indeed an effective implementation of it, then follow-up research can proceed in three directions: first, novel applications of this new technology, second, theoretical justifications, and finally, technical improvements to it. That is exactly what we experienced during the 1990s.

For the first line of research, Jasmina Arifovic, a student of Sargent’s, finished the first PhD dissertation that applied GAs to macroeconomics in 1991. It was not until 1994, however, that she published her work as a journal article. [10] replaced the rational representative firm in the cobweb model with Holland’s adaptive firms, and demonstrated how the adaptation of firms, driven by market forces (natural selection), collectively make the market price converge to the rational-expectations equilibrium price. Since then, a series of her papers has been published in various journals with a range of new application areas, including inflation [11], exchange rates [12] and coordination games [15].

Santa Fe Institute (SFI) Economics

Although Holland introduced this powerful toolkit to economists, he did not conduct any economic research with this toolkit himself, except for some joint work with Brian Arthur. Holland and Arthur met in September 1987 at a physics and economics Workshop hosted by the Santa Fe Institute. They had a great conversation on the nature of economics. The *chess* analogy proposed by Arthur led Holland to believe that the real problem with economics is “how do we make a science out of imperfectly smart agents exploring their way into an essentially infinite space of possibilities?” [152]: 151. On the other hand, Arthur was impressed by Holland’s approach to complex adaptive systems. Holland’s ideas of adaptation, emergence, and perpetual novelty, along with other notions, offered illuminating revelations to Arthur – insights he could never have had gained if he had confined himself to theorizing on equilibria.

This new vision of economics turned out to be the approach of the Santa Fe Institute when it established its economics program in 1988. The essence of the SFI economics was well documented by [19]. Instead of explaining genetic algorithms and classifier systems, which [92] had already done, this paper put a great emphasis on motivation. Arthur eloquently argued why the deductive approach should give way to the inductive approach when we are dealing with a model of heterogeneous agents. His paper thus built the microfoundation of economics upon agents' cognitive processes, such as pattern recognition, concept formation, and hypothesis formulation and refutation. Arthur then showed how the dynamics of these cognitive processes can be amenable to analysis with Holland's toolkit.

Maybe the best project to exemplify the SFI approach to economics is the *artificial stock market.* This research project started in 1988. Despite progress made in 1989, journal articles documenting this research were not available until 1994. [127] first built their stock market from a standard asset pricing model [89]. They then replaced the rational representative agent in the model with Holland's artificial adaptive agents, and then simulated the market.

For Arthur, the relevance of genetic algorithms to economics is much more than just strengthening the rational expectations equilibrium. He would like to see how one can use this tool to simulate the evolution of a real economy, such as the emergence of barter trading, money, a central bank, labor unions, and even Communism. However, he understood that one should start with a more modest problem than building a whole artificial economy, and this led to *the artificial stock market.*

Given this different motive, it is also interesting to see how SFI economists programmed agents in their models, and, given their coding or programming, how complex their agents can evolve to be. [127] also used the standard ternary string to code different types of trading rules frequently used by financial market traders. Each bit of a string was randomly drawn from the ternary alphabet $\{0, 1, *\}$. Each bit corresponds to the *condition part* of a single trading rule. For example, the condition part of a *double moving average rule* could be 'The 20-period moving average of price is above the 100-period moving average.' The appropriate bit is 1 if the condition is true, and 0 if it is false. They typically used strings of 70-80 symbols – that is, the same as the number of trading rules. This defines a search space of between 3^{70} and 3^{80} possible non-redundant classifiers. However, each artificial trader has *only* 60 classifiers in their own classifier system. Consider a case with 100 computerized traders: there are at most 6000 different rules being evaluated in one single trading run. Compared with the size of the search space, the number of rules is infinitesimal.

This rather large search space is certainly beyond what [19] called the *problem complex boundary*, a boundary beyond which arriving at the deductive solution and calculating it are unlikely or impossible for human agents, and this is where the SFI stock market comes into play. It provides the right place to use genetic algorithms and a great opportunity to watch evolution. As depicted by [19]: 24

> "We find no evidence that market behavior ever settles down; the population of predictors continually co-evolves. One way to test this is to take agents out of the system and inject them in again later on.

> If market behavior is stationary they should be able to do as well in the future as they are doing today. But we find that when we 'freeze' a successful agent's predictors early on and inject the agent into the system much later, the formerly successful agent is now a dinosaur. His predictions are unadapted and perform poorly. The system has changed. From our vantage point looking in, the market – the 'only game in town' on our computer – looks much the same. But internally it co-evolves and changes and transforms. It never settles."

Maybe the real issue is not whether GAs are used to strengthen the idea of REE, or to simulate artificial life, but *how we program adaptive agents*. This is crucial because different programming schemes may lead to different results. As Hahn pointed out, while there is only one way to be perfectly rational, there are an infinite number of ways to be partially rational ([152]: 250–251). This unlimited 'degree of freedom' of programming adaptive agents was also noticed by [132]: 2

> "This area is wilderness because the researcher faces so many choices after he decides to forgo the discipline provided by equilibrium theorizing."

Arthur would consider letting the agents start off 'perfectly stupid', and become smarter and smarter as they learn from experience. Now comes the core of the issue: how to program agents so that they can be initialized as perfectly stupid individuals, but can potentially get very smart. To answer this question, let us go back to the origin of genetic algorithms.

List Programming (LISP)

It is interesting to note that the binary strings initiated by Holland were originally motivated by an analogy to machine codes. After decoding, they can be computer programs written in a specific language, say, LISP or FORTRAN. Therefore, when a GA is used to evolve a population of binary strings, it behaves as if it is used to evolve a population of computer programs. If a decision rule is explicit enough not to cause any confusion in implementation, then one should be able to write it in a computer program. It is the *population of computer programs* (or their machine codes) which provides the most general representation of the *population of decision rules*. However, the equivalence between computer programs and machine codes *breaks down* when what is coded is the parameters of decision rules rather than decision rules (programs) themselves, as we often see in economic applications with GAs. The original meaning of evolving binary strings as evolving computer programs is lost.

The gradual loss of the original function of GAs has finally been noticed by Koza [104]. He chose the language LISP as the medium for the programs created by genetic programming (GP) because the syntax of LISP allows computer programs to be manipulated easily like the bit strings in GAs, so that the same genetic operations used on bit strings in GAs can also be applied to GP.

LISP S-expressions consist of either *atoms* or *lists*. Atoms are either members of a *terminal set*, that comprise the data (for example, constants and variables) to be used in the computer programs, or members of a *function set* that consists of a number of pre-specified functions or operators that are capable of processing any data value from the terminal set *and* any data value that results from the application of any function or operator in the function set. Lists are collections of atoms or lists, grouped within parentheses. In the LISP language, everything is expressed in terms of operators operating on some operands. The operator appears as the leftmost element in the parentheses and is followed by its operands and a closing (right) parenthesis. For example, the S-expression $(+X\ 3)$ consists of three atoms: from the left-most to right-most they are the function '+', the variable X and the constant 3. As another example, $(\times X\ (-Y\ 3))$ consists of two atoms and a list. The two atoms are the function '×' and the variable 'X,' which is then followed by the list $(-Y\ 3)$.

LISP was invented in the late 1950s by John McCarthy at MIT as a formalism for reasoning about the use of certain kinds of logical expressions, called recursion equations. LISP possesses unique features that make it an excellent medium for complex compositions of functions of various types, handling hierarchies, recursion, logical functions, self-modifying computer programs, self-executing computer programs, iterations, and structures whose size and shapes are dynamically determined. The most significant of these features is the fact that LISP descriptions of processes (routines) can themselves be represented and manipulated as LISP data (subroutines). As Koza demonstrated, LISP's flexibility in handling procedures as data makes it one of the most convenient languages in existence for exploring the idea of evolving computer programs genetically [104]. However, Koza and others have noted that the use of LISP is not necessary for genetic programming; what *is* important for genetic programming is the implementation of a LISP-like environment, where individual expressions can be manipulated like data, and are immediately executable.

Symbolic Regression

The distinguishing feature of GP is manifested by its first type of application in economics, known as *symbolic regression*. In symbolic regression, GP is used to discover the underlying data-generation process of a series of observations. While this type of application is well known to econometricians, the perspective from GP is novel. As Koza stated,

> "An important problem in economics is finding the mathematical relationship between the empirically observed variables measuring a system. In many conventional modeling techniques, one necessarily begins by selecting the size and shape of the model. After making this choice, one usually then tries to find the values of certain coefficients required by the particular model so as to achieve the best fit between the observed data and the model. But, in many cases, *the most important issue is the size and shape of the model itself.*" [105]: 57 (italics added)

Econometricians offer no general solution to the determination of size and shape (the functional form), but for Koza, finding the functional form of the model can be viewed as *searching a space of possible computer programs* for the particular computer program which produces the desired output for given inputs.

Koza employed GP to rediscover some basic physical laws from experimental data, for example, Kepler's third law and Ohm's law [104]. He then also applied it to eliciting a very fundamental economic law, namely, the *quantity theory of money* or the *exchange equation* [105]. Genetic programming was thus formally demonstrated as a *knowledge discovery* tool. This was probably the closest step ever made toward the original motivation of Holland's invention: 'Instead of trying to write your programs to perform a task you don't quite know how to do, *evolve them.*' Indeed, Koza did not evolve the parameters of an arbitrary chosen equation; instead, he evolved the whole equation from scratch. This style of application provides an evolutionary determination of bounded rationality.

Koza motivated a series of economic applications of genetic programming in the mid-1990s [105]. Chen and Yeh applied genetic programming to rediscovering *the efficient market hypothesis* in a financial time series [52]. They then moved one step forward to propose an alternative formulation of the efficient market hypothesis in the spirit of the *Kolmogorov complexity* of algorithms for pattern extraction from asset price data [54]. [54] and [141] employed GP to discover the underlying chaotic laws of motion of time series data. [6] and [123] also adopted a GP approach to discover profitable technical trading rules for the foreign exchange market and the stock market, respectively. Another area in which GP was actively applied is *option pricing*. [61] used GP for hedging derivative securities. [98] showed that genetically determined formulas outperformed most frequently quoted analytical approximations in calculating the implied volatility based on the Black-Scholes model. [65] and [99] derived approximations for calculating option prices and showed that GP-models outperformed various other models presented in the literature.

Needless to say, one can expect many more applications of GP to the automatic discovery of economic and financial knowledge (automatic generation of economic and financial knowledge in terms of their computer-programmed representations). However, its significant contribution to economics should not be mistaken for a perfect solution to knowledge discovery, data mining, or, more generally, *function optimization.* In a nutshell, genetic programming should be used to grow *evolving hierarchies* of building blocks (subroutines) – the basic units of learning and information, from an immense space of subroutines. All evolution can do is look for improvements, not perfection. Holland believed that these evolving hierarchies are generic in adaptation, and can play a key role in understanding human learning and adaptive processes.

4 Agent-Based Economic Simulations with CI

In this Section, we shall review the applications of CI to ACE. Given the size limitations, it is impossible to give an exhaustive survey here. We can therefore only review a few selected areas which we consider most representative and characterize the early development of the literature. We shall give a macroscopic view of the literature in Sects. 5.1 and 5.2, and introduce some most recent developments, which point to the future research, in Sect. 5.3.

4.1 The Cobweb Model

The cobweb model is a familiar playground in which to investigate the effects of production decisions on price dynamics. In this model consumers base their decisions on the current market price, but producers decide how much to produce based on the past prices. Agricultural commodities serve as a good example of the cobweb model. This model plays an important role in macroeconomics, because it is the place in which the concept 'rational expectations' originated [121]. Moreover, it is also the first neo-classical macroeconomic prototype to which an agent-based computational approach was applied [10]. This Section will first briefly formulate the cobweb model and then review the work on agent-based modeling of the cobweb model.

Consider a competitive market composed of n firms which produce the same goods by employing the same technology and which face the same cost function described in Eqn. (72):

$$c_{i,t} = xq_{i,t} + \frac{1}{2}ynq_{i,t}^2 \tag{72}$$

where $q_{i,t}$ is the quantity supplied by firm i at time t, and x and y are the parameters of the cost function. Since at time $t-1$, the price of the goods at time t, P_t, is not available, the decision about optimal $q_{i,t}$ must be based on the expectation (forecast) of P_t – that is, $P^e_{i,t}$. Given $P^e_{i,t}$ and the cost function $c_{i,t}$, the expected profit of firm i at time t can be expressed as follows:

$$\pi^e_{i,t} = P^e_{i,t}q_{i,t} - c_{i,t} \tag{73}$$

Given $P^e_{i,t}$, $q_{i,t}$ is chosen at a level such that $\pi^e_{i,t}$ can be maximized and, according to the first-order condition, is given by

$$q_{i,t} = \frac{1}{yn}(P^e_{i,t} - x) \tag{74}$$

Once $q_{i,t}$ is decided, the aggregate supply of the goods at time t is fixed and P_t, which sets demand equal to supply, is determined by the demand function:

$$P_t = A - B \sum_{i=1}^{n} q_{i,t} \tag{75}$$

where A and B are parameters of the demand function.

Given P_t, the actual profit of firm i at time t is:

$$\pi_{i,t} = P_t q_{i,t} - c_{i,t} \tag{76}$$

The neo-classical analysis simplifies the cobweb model by assuming the homogeneity of market participants – in other words, a representative agent. In such a setting, it can be shown that the homogeneous rational expectations equilibrium price (P^*) and quantity (Q^*) are ([53]: 449):

$$P^*_t = \frac{Ay + Bx}{B + y}; \quad Q^*_t = \frac{A - x}{B + y} \tag{77}$$

CI in the Agent-Based Cobweb Model

The neo-classical analysis based on homogeneous agents provides us with a limited understanding of the price dynamics or price instability in a real market, since firms' expectations of the prices and the resultant production decisions in general must be heterogeneous. Using genetic algorithms to model the adaptive behavior of firms' production, Arifovic gave the first agent-based model of the cobweb model [10] . She applied two versions of GAs to this model. The basic GA involves three genetic operators: reproduction, crossover, and mutation. Arifovic found that in each simulation of the basic GA, individual quantities and prices exhibited fluctuations for its entire duration and did not result in convergence to the rational expectations equilibrium values, which is quite inconsistent with experimental results with human subjects.

Arifovic's second GA version – the *augmented GA* – includes the election operator in addition to reproduction, crossover, and mutation. The election operator involves two steps. First, crossover is performed. Second, the potential fitness of the newly-generated offspring is compared with the actual fitness values of its parents. Among the two offspring and two parents, the two highest fitness individuals are then chosen. The purpose of this operator is to overcome difficulties related to the way mutation influences the convergence process, because the election operator can bring the variance of the population rules to zero as the algorithm converges to the equilibrium values.

The results of the simulations show that the augmented GA converges to the rational expectations equilibrium values for all sets of cobweb model parameter values, including both stable and unstable cases, and can capture several features of the experimental behavior of human subjects better than other simple learning algorithms. To avoid the arbitrariness of choice of an adaptive scheme, [114] suggested that comparison of the behavior of adaptive schemes with behavior observed in laboratory experiments with human subjects can facilitate the choice of a particular adaptive scheme. From this suggestion, the GA could be considered an appropriate choice to model learning agents in a complex system.

The application of genetic programming to the cobweb model started from [53], who compared the learning performance of GP-based learning agents with that of GA-based learning agents. They found that, like GA-based learning agents, GP-based learning agents also can learn the homogeneous rational expectations equilibrium price under both the stable and unstable cobweb case. However, the phenomenon of 'price euphoria', which did not happen in [10], does show up quite often at the early stages of the GP experiments. This is mainly because agents in their setup were initially endowed with very limited information as compared to [10]. Nevertheless, GP-based learning can quickly coordinate agents' beliefs so that the emergence of price euphoria is only temporary. Furthermore, unlike [10], Chen and Yeh did not use the election operator. Without the election operator, the rational expectations equilibrium is exposed to potentially persistent perturbations due to agents' adoption of the new, but untested, rules. However, what shows up in [53] is that the market can still bring any price deviation back to equilibrium. Therefore, the self-stabilizing feature of the market, known as the 'invisible hand', is more powerfully replicated in their GP-based artificial market.

The self-stabilizing feature of the market demonstrated in [53] was furthered tested with two complications. In the first case, [55] introduced a population of speculators to the market and examined the effect of speculations on market stability. In the second case, the market was perturbed with a structural change characterized by a shift in the demand curve; [57] then tested whether the market could restore the rational expectations equilibrium. The answer to the first experiment is generally negative, namely that speculators do not enhance the stability of the market; on the contrary, they destabilize the market. Only in special cases when trading regulations – such as the transaction cost and position limit – were tightly imposed could speculators enhance the market stability. The answer for the second experiment is, however, positive. Chen and Yeh showed that GP-based adaptive agents could detect the shift in the demand curve and adapt to it [57]. Nonetheless, the transition phase was non-linear and non-smooth; one can observe slumps, crashes, and bursts in the transition phase. In addition, the transition speed is uncertain. It could be fast, but could be slow as well.

This series of studies on the cobweb model enriches our understanding of the self-stabilizing feature of the market. The market has its limit, beyond which it can become unstable with crazy fluctuations. However, imposing trading regulations may relax the limit and enhance market stability. One is still curious to know where the self-stabilizing capability comes from in the first place. Economists have known for a long time that it comes from the free competition principle, or the survival-of-the-fittest principle. In GA or GP, this principle is implemented through *selection pressure*. Chen studied the role of selection pressure by replacing the usual proportionate selection scheme with the one based on the approximate uniform distribution, showing that if selection pressure is removed or alleviated, then the self-stabilizing feature is lost [37]. In a word, selection pressure plays the role of the invisible hand in economics.

It is interesting to know whether the time series data generated by the artificial market can replicate some dynamic properties observed in the real market. [46] and [57] started the analysis of the time series data generated from the artificial market. The time series data employed was generated by simulating the agent-based cobweb model with the presence of speculators. It was found that many stylized features well documented in financial econometrics can in principle be replicated from GP-based artificial markets, which include leptokutosis, non-IIDness, and volatility clustering. Furthermore, [57] performed a CUSUMSQ test, a statistical test for structural change, on the data. The test indicated the presence of structural changes in the data, which suggested that the complex interaction process of these GP-based producers and speculators can even generate endogenous structural changes.

4.2 Overlapping Generations Models

While there are several approaches to introducing dynamic general equilibrium structures to economics, the overlapping generations model (hereafter, OLG) may be regarded as the most popular one in current macroeconomics. Over the last two decades, the OLG model has been extensively applied to studies of savings, bequests, demand for assets, prices of assets, inflation, business cycles, economic growth, and the effects of taxes, social security, and budget deficits. In the following, we shall first give a brief illustration of a simple OLG model of inflation, a *two-period* OLG model.

Two-Period OLG Model

A simple OLG model can be described as follows. It consists of overlapping generations of two-period-lived agents. At time t, N young agents are born. Each of them lives for two periods $(t, t+1)$. At time t, each of them is endowed with e^1 units of a perishable consumption good, and with e^2 units at time $t+1$ ($e^1 > e^2 > 0$). Presumably e^1 is assumed to be greater than e^2 in order to increase the likelihood (but not ensure) that agents will choose to hold money from period 1 to 2 so as to push value forward. An agent born at time t consumes in both periods. Term c_t^1 is the consumption in the first period (t), and c_t^2 the second period ($t+1$). All agents have identical preference given by

$$U(c_t^1, c_t^2) = \ln(c_t^1) + \ln(c_t^2) \tag{78}$$

In addition to perishable consumption goods, there is an asset called *money* circulating in the society. The nominal money supply at time t, denoted by H_t, is exogenously determined by the government and is held distributively by the old generation at time t. For convenience, we shall define h_t to be $\frac{H_t}{N}$ – in other words, the nominal per capita money supply.

This simple OLG gives rise to the following agent's maximization problem at time t:

$$\max_{(c_{i,t}^1, c_{i,t}^2)} \quad \ln(c_{i,t}^1) + \ln(c_{i,t}^2)$$
$$\text{such that} \quad c_{i,t}^1 + \frac{m_{i,t}}{P_t} = e^1, \quad c_{i,t}^2 = e^2 + \frac{m_{i,t}}{P_{t+1}} \tag{79}$$

where $m_{i,t}$ represents the nominal money balances that agent i acquires at time period t and spends in time period $t+1$, and P_t denotes the nominal price level at time period t. Since P_{t+1} is not available at period t, what agents actually can do is to maximize their expected utility $E(U(c_t^1, c_t^2))$ by regarding P_{t+1} as a random variable, where $E(.)$ is the expectation operator. Because of the special nature of the utility function and budget constraints, the first-order conditions for this expected utility maximization problem reduce to the certainty equivalence form:

$$c_{i,t}^1 = \frac{1}{2}(e^1 + e^2 \pi_{i,t+1}^e) \tag{80}$$

where $\pi_{i,t+1}^e$ is agent i's expectation of the inflation rate $\pi_{t+1} (\equiv \frac{P_{t+1}}{P_t})$. This solution tells us the optimal decision of savings for agent i given her expectation of the inflation rate, $\pi_{i,t+1}^e$.

Suppose the government deficit G_t is all financed through seignorage and is constant over time ($G_t = G$). We can then derive the dynamics (time series) of nominal price $\{P_t\}$ and inflation rate $\{\pi_t\}$ from Eqn. (80). To see this, let us denote the savings of agent i at time t by $s_{i,t}$. Clearly,

$$s_{i,t} = e^1 - c_{i,t}^1 \tag{81}$$

From Eqn.(79), we know that

$$m_{i,t} = s_{i,t} P_t, \quad \forall i, t \tag{82}$$

In equilibrium, the nominal aggregate money demand must equal nominal money supply, namely,

$$\sum_{i=1}^{N} m_{i,t} = H_t = H_{t-1} + GP_t, \quad \forall t \tag{83}$$

The second equality says that the money supply at period t is the sum of the money supply at period $t-1$ and the nominal deficit at period t, GP_t. This equality holds, because we assume the government deficits are all financed by seignorage.

Summarizing Eqns. (82) and (83), we get

$$\sum_{i=1}^{N} s_{i,t} P_t = \sum_{i=1}^{N} s_{i,t-1} P_{t-1} + GP_t \tag{84}$$

The price dynamics are hence governed by the following equation:

$$\pi_t = \frac{P_t}{P_{t-1}} = \frac{\sum_{i=1}^{N} s_{i,t-1}}{\sum_{i=1}^{N} s_{i,t} - G} \tag{85}$$

Now suppose that each agent has perfect foresight, that is,

$$\pi^e_{i,t} = \pi_t, \quad \forall i, t \tag{86}$$

By substituting the first-order condition Eqn. (80) into Eqn. (84), the paths of equilibrium inflation rates under perfect foresight dynamics are then

$$\pi_{t+1} = \frac{e^1}{e^2} + 1 - \frac{2g}{e^2} - (\frac{e^1}{e^2})(\frac{1}{\pi_t}) \tag{87}$$

where $g = \frac{G}{N}$ is the real per capita deficit.

At steady state ($\pi_{t+1} = \pi_t$), Eqn. (87) has two real stationary solutions (fixed points), a low-inflation stationary equilibrium, π^*_L, and a high-inflation one, π^*_H, given by

$$\pi^*_L = \frac{1 + \frac{e^1}{e^2} - \frac{2g}{e^2} - \sqrt{(1 + \frac{e^1}{e^2} - \frac{2g}{e^2}) - 4\frac{e^1}{e^2}}}{2} \tag{88}$$

$$\pi^*_H = \frac{1 + \frac{e^1}{e^2} - \frac{2g}{e^2} + \sqrt{(1 + \frac{e^1}{e^2} - \frac{2g}{e^2}) - 4\frac{e^1}{e^2}}}{2} \tag{89}$$

Despite its popularity, the OLG models are well known for their multiplicity of equilibria, in our case, the coexistence of two inflation equilibria: Eqns. (88) and (89). Things can be even more intriguing if these equilibria have different welfare implications. In our case, the one with a higher inflation rate is the Pareto-inferior equilibrium, whereas the one with a lower inflation rate is the Pareto-superior equilibrium.

CI in Agent-Based OLG Models of Inflation

To see whether decentralized agents are able to coordinate intelligently to single out a Pareto-superior equilibrium rather than be trapped in a Pareto-inferior equilibrium, [11] proposed the first agent-based modification of an OLG model of inflation. She applied genetic algorithms (GAs) towards modeling the learning and adaptive behavior of households. In her study, GA-based agents were shown to be able to select the Pareto-superior equilibrium. She further compared the simulation results based on GAs with those from laboratories with human subjects, concluding that GAs were superior to other learning schemes, such as the recursive least squares.

This line of research was further carried out in [27,31–33] and [69]. Bullard and Duffy made the distinction between two implementations of GA learning: depending on what to encode, GA learning can be implemented in two different ways, namely, learning how to optimize [11] and learning how to forecast [33]. It was found that these two implementations lead to the same result: agents can indeed learn the Pareto-superior equilibrium. The only difference is the speed of convergence. The 'learning how to forecast' version of genetic algorithm learning converges faster than the 'learning how to optimize' implementation studied by [11]. Nevertheless, a robust analysis showed that coordination was more difficult when the number of inflation values considered (search space) by agents was higher, when government deficits increased, and when agents entertained inflation rate forecasts outside the bounds of possible stationary equilibria.

Chen and Yeh generalized Bullard and Duffy's 'learning how to forecast' version of GA learning with GP [56]. In [33], what agents learn is just a the inflation rate *per se*, rather than regularity about its *motion*, which is a function. Chen and Yeh considered it too restrictive to learn just a number. From [86], if the equilibrium of an OLG is characterized by limit cycles or strange attractors rather than by fixed points, then what agents need to learn is not just a number, but a functional relationship, such as $x_t = f(x_{t-1}, x_{t-2}, ...)$. Chen and Yeh therefore generalized Bullard and Duffy's evolution of *beliefs* from a sequence of populations of numbers to a sequence of populations of functions. Genetic programming serves as a convenient tool to make this extension.

The basic result observed in [56] is largely consistent with [10] and [33], namely, agents being able to coordinate their actions to achieve the Pareto-superior equilibrium. Furthermore, their experiments showed that the convergence is not sensitive to the initial rates of inflation. Hence, the Pareto-superior equilibrium has a large domain of attraction. A test on a structural change (a change in deficit regime) was also conducted. It was found that GP-based agents were capable of converging very fast to the new low-inflationary stationary equilibrium after the new deficit regime was imposed. However, the basic result was not insensitive to the dropping of the survival-of-the-fittest principle. When that golden principle was not enforced, we experienced dramatic fluctuations of inflation and occasionally the appearance of super inflation. The agents were generally worse off.

Birchenhall and Lin provided perhaps the most extensive coverage of robustness checks ever seen in agent-based macroeconomic models [27]. Their work covers two different levels of GA designs: one is genetic operators, and the other is architecture. For the former, they consider different implementations of the four main GA operators – namely, selection, crossover, mutation, and election. For the latter, they consider a single-population GA (social learning) versus a multi-population GA (individual learning). They found that Bullard and Duffy's results are sensitive to two main factors: the election operator and architecture. Their experimental results in fact lend support to some early findings – for example, the significance of the election operator [10] and the different consequences of social learning and individual learning [151]. What is particularly interesting is that individual learning reduces the rate of convergence to the same belief. This is certainly an important finding, because most studies on the convergence of GAs to Pareto optimality are based on the social learning version.

Ballard and Duffy studied a more complicated version of the two-period OLG model, based on [86]. They consider the following utility function for the households [32],

$$U(c_t^1, c_t^2) = \frac{\ln(c_t^1)^{1-\rho_1}}{1-\rho_1} + \frac{\ln(c_t^2)^{1-\rho_2}}{1-\rho_2} \tag{90}$$

Under time-separable preferences and provided that the value of the coefficient of relative risk aversion for the old agent (ρ_2) is high enough and that of the young agents is low enough (ρ_1), [86] showed that stationary perfect-foresight equilibria also may exist in which the equilibrium dynamics are characterized either as *periodic* or *chaotic trajectories* for the inflation rate, and these complicated stationary equilibria are also Pareto optimal. To have these possibilities, they set ρ_2 equal to 2 and then increased the value of this preference parameter up to 16 by increments of 0.1, while fixed ρ_1 at 0.5 in all cases.

The forecast rule considered by Bullard and Duffy is to use the price level that was realized $k+1$ periods in the past as the forecast of next period's price level, namely,

$$P_{i,t}^e = P_{t-k-1}, \quad k \in [0, \bar{k}] \tag{91}$$

In their case, $\bar{k}$ was set to 256, which allows the agents to take actions consistent with a periodic equilibrium of an order as high as 256. Alternatively, agent i's forecast of the gross inflation factor between dates t and $t+1$ is given by

$$\pi^e_{i,t} = \frac{P_{t-k-1}}{P_{t-1}} \tag{92}$$

As usual, the lifetime utility function was chosen as the fitness function to evaluate the performance of a particular forecast rule. Instead of roulette wheel selection, tournament selection was applied to create the next generation.

It was found that the stationary equilibria on which agents coordinate were always relatively simple – either a steady state or a low-order cycle. For low values of ρ_2 (in particular, those below 4.2), they observed convergence to the monetary steady state in every experiment, which is the same prediction made by the limited backward perfect-foresight dynamics. As ρ_2 was increased further, the limiting backward perfect foresight dynamics displayed a bifurcation, with the monetary steady state losing stability and never regaining it for values of $\rho_2 \geq 4.2$. However, in their system with learning, the monetary steady state was always a limit point in at least 1 of the 10 experiments conducted for each different value of ρ_2. Also, for $\rho_2 \geq 4.2$, their system often converged, in at least one experiment, to a period-2 stationary equilibrium, even in cases in which that equilibrium, too, had lost its stability in the backward perfect-foresight dynamics.

It is difficult, however, for an economy comprised of optimizing agents with initial heterogeneous beliefs to coordinate on especially complicated stationary equilibria, such as the period-k cycles where $k \geq 3$. In particular, the period-3 cycle that is stable in the backward perfect-foresight dynamics for values $\rho_2 \geq 13$ was never observed in their computational experiments. Interesting enough, three is the last entry of *Sarkovskii's ordering*, whereas one, two and four are first few entries.

They also found that the time it took agents to achieve coordination tended to increase with the relative risk aversion of the old agents over a large portion of the parameter space. Usually, it was the case when the system converged to the period-2 cycle. Moreover, when cycles exist, the transient dynamics of their systems could display qualitatively complication dynamics for long periods of time before eventually to relatively simple, low-periodicity equilibria.

A related phenomenon to cyclical equilibria is *sunspot equilibria*. The sunspot variable is the variable which has no intrinsic influence on an economy – in other words, it has nothing to do with an economy's fundamentals. Sunspot equilibria exist if the sunspot variable can impact the economy simply because a proportion of agents believe so and act accordingly to their belief. [22] showed that the connection between cyclical and sunspot equilibria is very close. They proved that a two-state stationary sunspot equilibrium exists if and only if a period-2 equilibrium exists. [69] started with an OLG model of inflation comparable to [32]. He studied an economy whose households have the following utility function,

$$U(c_t^1, c_t^2) = 0.1[c_t^1]^{0.9} + 10 - [\frac{10}{1 + c_t^2}]^2 \tag{93}$$

This utility function has the property that the concavity with respect to c_t^1 is much smaller than the concavity with respect to c_t^2, which is necessary for the existence of a periodic equilibrium [86].

He first found that in cases where periodic equilibria exist, households' beliefs were successfully coordinated to the period-2 cycle rather than the steady state. He then assumed all households to be sunspot believers and showed that households' beliefs converged to the sunspot equilibrium. In that case, the observed values of the price levels are completely governed by something which has nothing to do with the economy's fundamentals. Finally, he relaxed the assumption by simulating an explicit contest between 'sunspot believers' and 'sunspot agnostics'. The simulation showed that in most cases, the population consisted, after a rather short period, only of households whose actions depended on the value of the sunspot variable.

4.3 Foreign Exchange Rate Fluctuations

Another popular class of OLG models to which an agent-based approach is applied is the the OLG model of foreign exchange rates, which is a version of the two-country OLG model with fiat money [96].

The OLG Model of Exchange Rate

There are two countries in the model. The residents of both countries are identical in terms of their preferences and lifetime endowments. The basic description of each country is the same as the single-country OLG model. Each household of generation t is is endowed with e^1 units of a perishable consumption good at time t, and e^2 of the good at time $t+1$, and consumes c_t^1 of the consumption good when young and c_t^2 when old. Households in both countries have common preferences given by

$$U(c_t^1, c_t^2) = \ln(c_t^1) + \ln(c_t^2). \tag{94}$$

The government of each country issues its own unbacked currency, $H_{1,t}$ and $H_{2,t}$. Households can save only through acquiring these two currencies. There are no legal restrictions on holdings of foreign currency. Thus, the residents of both countries can freely hold both currencies in their portfolios. A household at generation t solves the following optimization problem at time t:

$$\begin{gathered} \max_{(c_{i,t}^1, m_{i,1,t})} \quad \ln(c_{i,t}^1) + \ln(c_{i,t}^2) \\ \text{such that} \quad c_{i,t}^1 + \frac{m_{i,1,t}}{P_{1,t}} + \frac{m_{i,2,t}}{P_{2,t}} = e^1, \quad c_{i,t}^2 = e^2 + \frac{m_{i,1,t}}{P_{1,t+1}} + \frac{m_{i,2,t}}{P_{2,t+1}} \end{gathered} \tag{95}$$

where $m_{i,1,t}$ is household i' nominal holdings of currency 1 acquired at time t, $m_{i,2,t}$ is household i' nominal holdings of currency 2 acquired at time t, $P_{1,t}$ is the nominal price of the good in terms of currency 1 at time t, and $P_{2,t}$ is the nominal price of the good in terms of currency 2 at time t. The savings of household i at time t by $s_{i,t}$ is

$$s_{i,t} = e^1 - c^1_{i,t} = \frac{m_{i,1,t}}{P_{1,t}} + \frac{m_{i,2,t}}{P_{2,t}} \tag{96}$$

The exchange rate e_t between the two currencies is defined as $e_t = P_{1,t}/P_{2,t}$. When there is no uncertainty, the return on the two currencies must be equal,

$$R_t = R_{1,t} = R_{2,t} = \frac{P_{1,t}}{P_{1,t+1}} = \frac{P_{2,t}}{P_{2,t+1}}, \quad t \geq 1 \tag{97}$$

where $R_{1,t}$ and $R_{2,t}$ are the gross real rate of return between t and $t+1$, respectively. Rearranging Eqn. (97), we obtain

$$\frac{P_{1,t+1}}{P_{2,t+1}} = \frac{P_{1,t}}{P_{2,t}} \quad t \geq 1 \tag{98}$$

From Eqn. (98) it follows that the exchange rate is constant over time:

$$e_{t+1} = e_t = e, \quad t \geq 1 \tag{99}$$

Savings demand derived from household's maximization problem is given by

$$s_{i,t} = \frac{m_{i,1,t}}{p_{1,t}} + \frac{m_{i,2,t}}{p_{2,t}} = \frac{1}{2}[e^1 - e^2\frac{1}{R_t}] \tag{100}$$

Aggregate savings of the world at time period t, S_t, are equal to the sum of their savings in terms of currency 1, $S_{1,t}$, and in terms of currency 2, $S_{2,t}$. With the homogeneity assumption, we have

$$S_{1,t} = \sum_{i=1}^{2N} \frac{m_{i,1,t}}{P_{1,t}} = \frac{2Nm_{1,t}}{P_{1,t}} \tag{101}$$

and

$$S_{2,t} = \sum_{i=1}^{2N} \frac{m_{i,2,t}}{P_{2,t}} = \frac{2Nm_{2,t}}{P_{2,t}} \tag{102}$$

The equilibrium condition in the loan market requires

$$S_t = S_{1,t} + S_{2,t} = N[e^1 - e^2\frac{P_{1,t+1}}{P_{1,t}}] = \frac{H_{1,t} + H_{2,t}e}{P_{1,t}} \tag{103}$$

Eqn. (103) only informs us of the real saving in terms of the real world money demand. This equation alone cannot determine the household real demands for each currency. Hence, this equation cannot uniquely determine a set of price $(P_{1,t}, P_{2,t})$, and leave the exchange rate indeterminate as well. This is known as the famous *indeterminacy of exchange rate proposition.* The proposition says that if there exists a monetary equilibrium in which both currencies are valued at some exchange rate e, then there exists a monetary equilibrium at any exchange rate $\hat{e} \in (0, \infty)$ associated with a different price sequence $\{\hat{P}_{1,t}, \hat{P}_{2,t}\}$ such that

$$R_t = \frac{P_{1,t}}{P_{1,t+1}} = \frac{P_{2,t}}{P_{2,t+1}} = \frac{\hat{P}_{1,t}}{\hat{P}_{1,t+1}} = \frac{\hat{P}_{2,t}}{\hat{P}_{2,t+1}} \tag{104}$$

and

$$S_t = \frac{H_{1,t} + H_{2,t}e}{P_{1,t}} = \frac{H_{1,t} + H_{2,t}\hat{e}}{\hat{P}_{1,t}} \tag{105}$$

where

$$\hat{P}_{1,t} = \frac{H_{1,t} + \hat{e}H_{2,t}P_{1,t}}{H_{1,t} + eH_{2,t}}, \quad \hat{P}_{2,t} = \frac{\hat{P}_{1,t}}{\hat{e}}. \tag{106}$$

Rearranging Eqn. (103), one can derive the law of motion of $P_{1,t}$

$$P_{1,t+1} = \frac{e^1}{e^2}P_{1,t} - \frac{H_{1,t} + eH_{2,t}}{Ne^2} \tag{107}$$

For any given exchange rate e, this economy with constant supplies of both currencies, H_1 and H_2, has a steady-state equilibrium, namely,

$$P_{1,t+1} = P_{1,t} = P_1^* = \frac{H_1 + eH_2}{N(e^1 - e^2)} \tag{108}$$

Like e, the level of P_1^* is also indeterminate. In addition, since households are indifferent between the currencies that have the same rates of return in the homogeneous-expectations equilibrium, the OLG model in which agents are rational does not provide a way to determine the portfolio $\lambda_{i,t}$, which is the fraction of the savings placed into currency 1.

CI in Agent-Based OLG Models of the Exchange Rate

In order to examine the behavior of the exchange rate and the associated price dynamics, Arifovic initiated the agent-based modeling of the exchange rate in the context of the OLG model [12]. In the OLG model of the exchange rate, households have two decisions to make when they are young, namely, saving $(s_{i,t})$ and portfolio $(\lambda_{i,t})$. These two decisions were encoded by concatenation of two binary strings, the first of which encoded $s_{i,t}$, whereas the second of which encoded $\lambda_{i,t}$. The single-population augmented genetic algorithm was then applied to evolve these decision rules. The length of a binary string, l, is 30: The first 20 elements of a string encode the first-period consumption of agent i of generation t; the remaining 10 elements encode the portfolio fraction of agent i:

$$\underbrace{010100...110}_{20 \quad bits: s_{i,t}} \; \underbrace{101..001}_{10 \quad bits: \lambda_{i,t}}$$

While Eqn. (99) predicts the constancy of the exchange rate, genetic algorithm simulations conducted by [12] indicated no sign of the setting of the exchange rate to a constant value. Instead, they showed persistent fluctuations of the exchange rate. Adaptive economic agents in this model can, in effect, endogenously generate *self-fulfilling arbitrage opportunities*, which in turn make exchange rates continuously fluctuate.

The fluctuating exchange rate was further examined using formal statistical tests in both [12] and [16]. First, in [12], the stationarity (Dickey-Fuller) test was applied to examine whether the exchange rate series is non-stationary. The result of the test did not indicate non-stationarity. Second, [16] analyzed the statistical properties of the exchange rate returns, namely, the logarithm of e_t/e_{e-1}. The independence tests (Ljung-BOx-Pierce and BDS) clearly rule out the lack of persistence (dependence) in the return series. Third, they plotted the phase diagrams of the return series and found that there is a well-defined attractor for all series. The shapes of the attractor are robust to the changes in the OLG model parameters as well as to the changes in the GA parameters. Fourth, to verify that this attractor is chaotic, the largest two Lyapunov exponents were calculated. The largest Lyapnov exponent is positive in all series, which supports that attractors under investigation are chaotic. Finally, volatility clustering was also found to be significant in the return series. This series of econometric examinations confirms that agent-based modeling is able to replicate some stylized facts known in financial markets.

Arifovic considered a different application of GAs to modeling the adaptive behavior of households [14]. Instead of savings and portfolio decision rules, she turned to the forecasting behavior of households. The forecasting models of exchange rates employed by agents are simple moving-average models. They differ in the rolling window size, which are endogenously determined and can be time-variant. What is encoded by GAs is the size of the rolling window rather than the usual savings and portfolio decision. Simulations with this new coding scheme resulted in the convergence of the economies to a single-currency equilibrium – that is, the collapse of one of the two currencies. This result was not found in [12]. This study therefore shows that different implementations of GA learning may have non-trivial effects on the simulation results. In one implementation, one can have persistent fluctuation of the exchange rate [12]; in another case, one can have a single-currency equilibrium.

Following the design of [81], Arifovic combined two different applications of GA learning. In addition to the original population of agents, who are learning how to forecast, she added another population of agents, who are learning how to optimize [14]. Nevertheless, unlike [81], these two populations of agents did not compete with each other. Instead, they underwent separate genetic algorithm updating. Simulations with these two separate evolving populations did not have the convergence to single currency equilibrium, but were characterized instead by persistent fluctuation.

A different scenario of the currency collapse is also shown in [13], which is an integration of the OLG model of exchange rate with the OLG model of inflation. In this model, the governments of both countries have constant deficits ($G_i, i = 1, 2$) which were financed via seignorage,

$$G_i = \frac{H_{i,t} - H_{i,t-1}}{P_{i,t}}, \quad i = 1, 2 \tag{109}$$

Combining Eqns. (103) and (109) gives the condition for the monetary equilibrium in which both governments finance their deficits via seignorage:

$$G_1 + G_2 = S_t - S_{t-1}R_{t-1} \tag{110}$$

This integrated model inherits the indeterminacy of the exchange rate from the OLG model of the exchange rate and the indeterminacy of the inflation rate from the OLG model of inflation. Any constant exchange rate e ($e \in (0, \infty)$) is an equilibrium that supports the same stream of government deficits (G_1, G_2), and the same equilibrium gross rate of return (and thus the same equilibrium savings). The existence of these equilibrium exchange rates indicates that the currencies of both countries are valued despite the difference of the two countries' deficits. In fact, in equilibrium the high-deficit country and the low-deficit county experience the same inflation rate, and hence so do their currencies' rates of return. Nonetheless, since the high-deficit country has a higher money supply, if both currencies are valued, then the currency of the high-deficit country will eventually drive the currency of the low-deficit country out of households' portfolios. Given this result, it might be in the interest of a country with lower deficits to impose a degree of capital control.

Arifovic showed that agent-based dynamics behave quite different from the above homogeneous rational expectations equilibrium analysis [13]. In her agent-based environment, the evolution of households' decision rules of savings and portfolio results in a flight away from the currency used to finance the larger of the two deficits. In the end, households hold all of their savings in the currency used to finance the lower of the deficits. Thus, the economy converges to the equilibrium in which only the low-deficit currency is valued. The currency of the country that finances the larger of the two deficits become valueless, and we have a single-currency equilibrium again.

4.4 Artificial Stock Markets

Among all applications of the agent-based approach to macroeconomic modeling, the most exciting one is the *artificial stock market*. By all standards, the stock market is qualified to be a complex adaptive system. However, conventional financial models are not capable of demonstrating this feature. On the contrary, the famous *no-trade theorem* shows how inactive this market can be in equilibrium [146]. It was therefore invigorating when Holland and Arthur established an economics program at the Santa Fe Institute in 1988 and chose artificial stock markets as their initial research project. The SFI artificial stock market is built upon the standard asset pricing model [88, 89]. What one can possibly learn from this novel approach was well summarized in [127], which is in fact the first journal publication on an agent-based artificial stock market. A series of follow-up studies materialized the content of this new fascinating frontier in finance.

Agent Engineering and Trading Mechanisms

Agent-based artificial stock markets have two mainstays: agent engineering and institution (trading mechanism) designs. Agent engineering mainly concerns the construction of financial agents. [144] showed how to use genetic algorithms to encode trading strategies of traders. A genetic fuzzy approach to modeling trader's behavior was shown in [143], whereas the genetic neural approach was taken by [110]. To simulate the agent-based artificial stock market based on the standard asset pricing model, the AI-ECON Research Center at the National Chengchi University, Taiwan developed software known as the *AI-ECON Artificial Stock Market (AIE-ASM)*. The *AIE Artificial Stock Market* differs from the *SFI Artificial Stock Market* in the computational tool that is employed. The former applies genetic programming, while the latter has genetic algorithms. In AIE-ASM, genetic programming is used to model agents' expectations of the price and dividends. A menu-like introduction to AIE-ASM Ver. 2 can be found in [63].

In [35] and [160] we see a perfect example of bringing different learning schemes into the model. The learning schemes incorporated into [35] include empirical Bayesian traders, momentum traders, and nearest-neighbor traders, whereas those included in [160] are neural network and momentum traders. [109] gave a more thorough and general discussion of the construction of artificial financial agents. In addition to models, data is another dimension of agent engineering. What can be addressed here is the issue of stationarity that the series traders are looking at. Is the entire time series representative of the same dynamic process, or have things changed in the recent past? LeBaron studied traders who are initially heterogeneous in perception with different time horizons, which characterize their interpretation of how much of the past is relevant to the current decision making [110].

Chen and Yeh contributed to agent engineering by proposing a modified version of social learning [58]. The idea is to include a mechanism, called the *business school*. Knowledge in the business school is open for everyone. Traders can visit the business school when they are under great survival pressure. The social learning version of genetic programming is applied to model the evolution of the business school rather than directly on traders. Doing it this way, one can avoid making an implausible assumption that trading strategies, as business secrets, are directly imitable. [161] further combined this modified social learning scheme with the conventional individual learning scheme in an integrated model. In this integrated model a more realistic description of traders' learning behavior is accomplished: the traders can choose to visit the business school (learning socially), to learn exclusively from their experience (learning individually), or both. In their experiments, based on the effectiveness of different learning schemes, traders will switch between social learning and individual learning. Allowing such a competition between these two learning styles, their experiment showed that it is the individual learning style which won the trust of the majority. To the best of our knowledge, this is the only study which leaves the choice of the two learning styles to be endogenously determined.

The second component of agent-based stock markets is the institutional design. An institutional design should answer the following five questions: (i) who can trade, (ii) when and how can orders be submitted, (iii) who may see or handle the orders, (iv) how are orders processed, and (v) how are prices eventually set. Trading institutional designs in the conventional SFI artificial stock market either follow the Walrasian 'tatonnement' scheme or the rationing scheme. This scheme describes a market operation procedure. Basically, there is an auctioneer who serves as a market coordinator. In each market period, the auctioneer announces a price to all market participants. Based on this market price, participants submit their transaction plans, for instance how much to buy or how much to sell. The auctioneer will then collect all submissions. If there is an imbalance between demand and supply, the auctioneer will then announce a new price, and the market participants will submit new plans accordingly. This process continues until the auctioneer finds a price which can equate demand to supply, and all transaction plans will be carried out with this price, also called the 'equilibrium price'. The essence of tatonnement is that no single transaction can be allowed unless the equilibrium price is found. This highly centralized trading system needs to be distinguished from other less centralized or distributed trading systems.

[35] and [160], however, considered a double auction mechanism. This design narrows the gap between artificial markets and the real market, and hence makes it possible to compare the simulation results with the behavior of real data, such as tick-by-tick data. Since stock market experiments with human subjects were also conducted within the double auction framework [139], this also facilitates conversation between the experimental stock market and the agent-based artificial stock market.

Based on agent engineering and trading mechanism designs, agent-based artificial stock markets can generate various market dynamics, including price, trading volumes, the heterogeneity and complexity of traders' behavior, and wealth distribution. Among them, price dynamics is the one under the most intensive study. This is not surprising, because ever since the 1960s price dynamics has been the focus of studies on random walks, the efficient market hypothesis, and market rationality (the rational expectations hypothesis). With the advancement of econometrics, it further became the focus of the study of non-linear dynamics in the 1980s.

Mis-Pricing

Agent-based artificial stock markets make two important contributions to our understanding of the behavior of stock prices. First, they enable us to understand what may cause the price to deviate from rational equilibrium price or the so-called 'fundamental value'.

Both [35] and [160] discussed the effect of momentum traders on price deviation. Yang found that the presence of momentum traders can drive the market price away from the homogeneous rational equilibrium price [160]. Chan reported a similar finding: adding momentum traders to a population of empirical Bayesian traders has an adverse impact on market performance, although price deviation decreased as time went on [35]. Empirical Bayesian basically behaves like a Bayesian, except that the posterior distribution is built upon the empirical rather upon a subjective distribution. For example, in this context, the empirical Bayesian trader forms its posterior distribution of the dividends by using the empirical distributions of both dividends and prices.

LeBaron inquired whether agents with a long-horizon perception can learn to effectively use their information to generate a relatively stable trading environment [110]. The experimental results indicated that while the simple model structure with fixed long horizon agents replicates the usual efficient market results, the route to evolving a population of short horizon agents to long horizons may be difficult. [20] and [111] found that when the speed of learning (the length of a genetic updating cycle) decreased (which forces agents to look at longer horizon features), the market approached the REE.

[47] is another study devoted to price deviation. They examined how well a population of financial agents can track the equilibrium price. By simulating the artificial stock market with different dividend processes, interest rates, risk attitudes, and market sizes, they found that the market price is not an unbiased estimator of the equilibrium price. Except in a few extremely bad cases, the market price deviates from the equilibrium price moderately from -4% to $+16\%$. The pricing errors are in fact not patternless. They are actually negatively related to market sizes: a thinner market size tends to have a larger pricing error, and a thicker market tends to have a smaller one. For the thickest market which they have simulated, the mean pricing error is only 2.17%. This figure suggests that the new classical simplification of a complex world may still provide a useful approximation if some conditions are met, such as in this case, the market size.

Complex Dynamics

As to the second contribution, agent-based artificial stock markets also enhance our understanding of several stylized features well documented in financial econometrics, such as fat tails, volatility clusters, and non-linear dependence. [111] showed that the appearance of the ARCH effect and the non-linear dependence can be related to the speed of learning. [160] found that the inclusion of momentum traders generates a lot of stylized features, such as excess volatility, excess kurtosis (leptokurtotic), lack of serial independence of return, and high trading volume.

Another interesting line is the study of emergent properties within the context of artificial stock markets. Emergence is about "how large interacting ensembles exhibit a collective behavior that is very different from anything one may have expected from simply scaling up the behavior of the individual units" ([107]: 3). Consider the efficient market hypothesis (EMH) as an example. If none of the traders believe in the EMH, then this property will not be expected to be a feature of their collective behavior. Thus, if the collective behavior of these traders indeed satisfies the EMH as tested by standard econometric procedures, then we would consider the EMH as an emergent property. As another example, consider the rational expectations hypothesis (REH). It would be an emergent property if all our traders are boundedly rational, with their collective behavior satisfying the REH as tested by econometrics.

Chen and Yeh applied a series of econometric tests to show that the EMH and the REH can be satisfied with some portions of the artificial time series [59]. However, by analyzing traders' behavior, they showed that these aggregate results cannot be interpreted as a simple scaling-up of individual behavior. The main feature that produces the emergent results may be attributed to the use of genetic programming, which allows us to generate a very large search space. This large space can potentially support many forecasting models in capturing short-term predictability, which makes simple beliefs (such as that where the dividend is an iid (independent and identically distributed) series, or that when the price follows a random walk) difficult to be accepted by traders. In addition to preventing traders from easily accepting simple beliefs, another consequence of a huge search space is the generation of sunspot-like signals through mutually-reinforcing expectations. Traders provided with a huge search space may look for something which is originally irrelevant to price forecasts. However, there is a chance that such kinds of attempts may mutually become reinforced and validated. The generation of sunspot-like signals will then drive traders further away from accepting simple beliefs.

Using Granger causality tests, [59] found that dividends indeed can help forecast returns. By their experimental design, the dividend does not contain the information of future returns. What happens is a typical case of mutually-supportive expectations that make the dividend eventually contain the information of future returns.

As demonstrated in [58] and [59], one of the advantages of agent-based computational economics (the bottom-up approach) is that it allows us to observe what traders are actually thinking and doing. Are they martingale believers? Are they sunspot believers? Do they believe that trading volume can help predict returns? By counting the number of traders who actually use sunspots or trading volumes to forecast returns, one can examine whether sunspot effects and the causal relation between stock returns and trading volume can be two other emergent properties [49, 62].

Market Diversity and Market Efficiency

Yeh and Chen examined another important aspect of agent engineering, this being *market size* (number of market participants) [162]. Few studies have addressed the significance of market size on the performance of agent-based artificial markets. One good exception is [26], whose simulation results showed that the simple tradable emission permit scheme (an auction scheme) can be the most effective means for pollution control when the number of participants is small. However, as the number of participants increases, its performance declines dramatically and becomes inferior to that of the uniform tax scheme. Another exception is [33]. In most studies, the number of market participants is usually determined in an arbitrary way, mainly constrained by the computational load. [10], however, justified the number of participants from the viewpoint of search efficiency. She mentioned that the minimal number of strings (agents) for an effective search is usually taken to be 30 according to the artificial intelligence literature. Nonetheless, agent-based artificial markets have different purposes and concerns.

Related to market size is *population size*. In the case of social learning (single-population GA or GP), market size is the same as population size. However, in the case of individual learning (multi-population GA or GP), population size refers to something different, namely, the number of solution candidates each trader has. Like market size, population size is also arbitrarily determined in practice.

Yeh and Chen studied the effect of market size and population size upon market efficiency and market diversity under social and individual learning styles [162] . Their experimental results can be summarized as two effects on market efficiency (price predictability), namely, the *size effect* and the *learning effect*. The size effect says that the market will become efficient when the number of traders (market size) and/or the number of models (GP trees) processed by each trader (population size) increases. The learning effect says that the price will become more efficient if traders' adaptive behavior becomes more independent and private. Taking a look at market diversity, we observe very similar effects except for population size: market diversity does not go up with population size. These findings motivate us to search for a linkage between market diversity and market efficiency. A 'theorem' may go as follows: a larger market size and a more independent learning style will increase the diversity of traders' expectations, which in turn make the market become more active (high trading volume), and hence more efficient (less predictable). Their simulation results on trading volumes also supported this 'theorem'. They further applied this 'theorem' to explain why the US stock market behaves more efficiently than Taiwan's stock market.

4.5 Market/Policy Design

One of the research areas in which agent-based computational economics and experimental economics are closely intertwined is the *double-auction market* (DA market), or the agent-based DA market. The agent-based market serves as a good starting point for applying agent-based simulation to market/policy design. One important application of agent-based computational models to market/policy design is the electricity supply market [29, 125, 126]. In this application area, we are convinced that agent engineering (learning schemes) plays a crucial role in simulating the consequences of various market designs.

By agent engineering, [73] categorized agent-based models which have been developed to characterize or understand data from human subject experiments into three classes, namely, zero intelligent (ZI) agents, reinforcement and belief learning, and evolutionary algorithms. Among the three, ZI agents were considered to be a useful benchmark or a good building block for developing more advanced agent-based models. Zero-intelligent agents are introduced by [85], which is the earliest ACE work motivated by the double-auction market experiment.[21] However, as far as market efficiency is concerned, ZI traders are not sufficient for the market to converge to the social-welfare maximization price, or the equilibrium price. The necessary condition, therefore, requires agents to learn. Among all learning agents studied in the agent-based DA models, the simplest one is the ZI Plus (ZIP) agents, introduced by [66].

Wu and Bhattacharyya continued this line of research and studied the boundary beyond which ZIP traders may fail the market mechanism [159]. They introduced speculators into standard DA markets. They found that ZIP traders can no longer guarantee market efficiency when there is a large number of speculators, as compared to the number of normal traders. In some scenarios, the efficiency losses about 25% of the social welfare.

The purpose in studying the agent-based double auction (DA) market is to adequately equip ourselves to tackle the much more complex agent-based electricity market. [124] gave a splendid review of the well-known Electricity Market Complex Adaptive System (EMCAS) developed by the Argonne

[21] For a survey of later developments, see [38].

National Laboratory. EMCAS is an agent-based electricity supply market model written using the Recursive Agent Simulation Toolkit (`Repast`), a special-purpose agent-based simulation tool. The research on the agent-based electricity market is motivated by the undergoing transition from centrally regulated electricity markets to decentralized markets. These transitions introduce a highly intricate web of interactions of a large number of heterogeneous companies and players, which causes the consequences of new regulatory structures largely unknown and leaves policy design in a state of high stakes.[22] Given this uncertainty, agent-based models can help construct suitable laboratories that can provide ranges of possibilities and test regulatory structures before they are actually implemented. EMCAS now serves as the basis for evaluating Illinois' deregulation of the market.

Boyle presented an ambitious project on the agent-based model of the whole criminal justice system in the UK, which was funded by the Home Office in the UK [30]. The criminal justice system in England is delivered by three diverse government bodies, the Home Office, the Department of Constitutional Affairs, and the Crown Prosecution Service. Within the criminal justice system as a whole, there must be some dependencies among the functions of the three agencies. Nonetheless, the three constituents might not have been 'joined up' sufficiently well to encourage the best use of resources, and this caught the attention of the Treasury in their biennial spending review. Therefore, the purpose of this project is to build an agent-based model to help diverse operating groups engage in strategic policy making and take into account the complex interactions within the criminal justice system so as to better observe the impacts of policy.

To make the model fulfill this objective, [30] introduced a new thinking regarding agent-based models, called *the mirror function*, which is equivalent to producing a model of the whole criminal justice system in which all actors in the system acknowledge that the model was really 'them'. The work entailed gathering evidence of links between the behavior and actions of one person or group of people, and those of another, and through this making arguments for the best use of resources, while also reaching agreement between each group of people regarding all of this. This is essentially to do with encouraging a change in the style of working of these core government agencies. Boyle therefore demonstrates a very distinctive class of agent-based models, which integrates a vein of social work into model-building [30].[23]

---

[22] This is exemplified by the extremely unsatisfactory experience of California. While, according to economic theory, deregulation and free competition will lead to increased economic efficiency expressed in higher quality services and products at lower prices, the reality of today's emerging electricity markets does not fit this straightforward economic model.

[23] At present, there are very few agent-based models of this sort; [120] is the only case known to this author.

5 Pushing the Research Frontier with CI

5.1 Developments in Agent Engineering

An essential element of the agent-based modeling is *agent engineering*. Over the last decade, the progress made in modeling adaptive behavior has been particularly noticeable. There seems to have been a general tendency to *enrich* agents' adaptive behavior from several different perspectives. This enrichment has been made possible mainly due to extensive applications of *computational intelligence* to economics.

First, simple adaptive behavior has been extended to complex adaptive behavior. Initially, agents' decisions were simply characterized by *parametric* models; usually, there were just numbers over a bounded real space. [10]–[12] are typical examples (see Sect. 4 for details). All important decisions such as quantity supply, labor supply, savings, financial portfolios, and investment in human capital were characterized by *numbers* rather than *rules*. As a result, the things revealed by the adaptive processes were best viewed as a series of *number crunching* exercises. Sophisticated learning or adaptive behavior were not able to appear in these simple adaptive models.

Later on, the notion of using *rules* instead of *numbers* to characterize agents' decisions was brought in by [31]– [33], [81], and many others. These series of efforts brought about discernible progress: they formally introduced agents *which are able to forecast with rules (models)*. Nonetheless, their forecasting behavior was largely confined to *linear regression models*. This restriction was unavoidable because at that stage economists did not know much about dealing with non-parametric adaptive behavior, and linear regression models seemed to be the natural starting point. However, there is neither sound theoretic nor empirical support for the assumption that agents' adaptive behavior may be parameterized.

A breakthrough was made by [9], [53] and [112] via genetic programming (GP). The use of genetic programming not only makes agents able to engage in non-linear and non-parametric forecasting, but it also makes them able to *think* and *reason*. This last virtue is crucial because it helps us to represent a larger class of cognitive capabilities, such as making plans and strategies. This development contributes to the advancement of the agent-based models which are full of the non-trivial strategic behavior of agents, for instance, games, auctions, and financial markets. The AI-ECON Research Center in Taipei has now launched a research project – referred to as *the Functional-Modularity Foundation of Economics* – that has further enlarged the adaptive behavior to encompass preferences, commodities, technology, human capital, and organizations [41, 42].

By manipulating a set of primitives with genetic operators, one can *grow* a great variety of human cognitive processes. In principle, there is no limit to those growing processes. It is the *survival pressure* endogenously generated via agent interactions that determines their size. In this case, neither do we need to assume that the agents follow simple rules, as the KISS (keep it simple, stupid) principle suggests, nor do we assume that they are sophisticated. Simplicity or complexity is not a matter of an *assumption* but a matter of *emergence*. For example, in a simple deterministic agent-based cobweb model, the literature shows that all surviving firms have indeed followed simple and myopic rules to forecast price. However, their behavior became more complicated when speculators were introduced into the markets. In addition, when turning to the stock market, agents' behavior could switch between simple rules and sophisticated rules.[24] In a nutshell, in ACE, what determines the survivability of a type of agent is not the model designers, but the *natural law* of the models; we shall see more on this in Sect. 5.2.

The second development is concerned with the *behavioral foundations* of agent engineering. While CI tools have been extensively applied to agent engineering, their ability to represent *sensible* adaptive behavior has been questioned since agent-based economic models became popular. Since 1999, a series of efforts have been made in an attempt to justify the use of genetic algorithms in agent-based modeling. However, most of these studies are mainly built upon theoretical arguments. [94] were the first to use evidence from *interviews* and *questionnaires* to justify the use of genetic algorithms in their agent-based foreign exchange markets. Their study highlights the significance of the *field study* – an approach frequently used by sociologists – to agent engineering. Duffy's agent-based model of a medium of exchange applied the data from laboratory experiments with human subjects to justify the use of *reinforcement learning* [72]. His study showed how agent-based economic models can benefit from *experimental economics*.

Another related development has occurred in the use of *natural language*. People frequently and routinely use natural language or linguistic values, such as 'high', 'low', and so on, to describe their perception, demands, expectations, and decisions. Some psychologists have argued that our ability to process information efficiently is the outcome of applying *fuzzy logic* as part of our thought process. Evidence on human reasoning and human thought processes supports the hypothesis that at least some categories of human thought are definitely fuzzy. Yet, early agent-based economic models have assumed that an agent's adaptive behavior is *crisp*. Tay and Linn made progress in this direction by using a *genetic-fuzzy classifier system* (GFCS) to model traders' adaptive behavior in an artificial stock market [143].

[143] provided a good illustration of the *non-equivalence* between the acknowledgement of the *cognitive constraint* and the assumption of *simple agents*. It is well-known that the human mind is notoriously bad at intuitively

[24] In plain English parlance, they sometimes regarded George Soros as their hero, while at other times they developed a great admiration for Warren Buffett.

comprehending exponential growth. However, there is no evidence that traders on Wall Street are simple-minded. Tay and Linn's work recognized the difference, and appropriately applied the GFCS to *lessen* agents' reasoning load via the use of natural language.

[72], [94], and [143] can all be regarded as a starting point for a more remarkable development in agent engineering: the CI tools employed to model agents' adaptive behavior are grounded in strong evidence within the cognitive sciences. It is at this point that agent-based modeling should have closer interactions with the *field and panel study*, *experimental economics* and *behavioral economics* (See more below in Sect. 5.3).

5.2 Distinguishing Features

While the progress made in agent engineering is evident, a more subtle issue of ACE is: "does agent-based computational economics have anything worthwhile to offer economists in general, or is it only of interest to practitioners of its own paradigm?" In this Section, we shall argue that the development of ACE has already demonstrated some distinguishing features with insightful lessons which are generally not available from neoclassical macroeconomics. The distinguishing features, which may interest economists in general, are two-fold. First, ACE helps build a *true* micro-foundation of macroeconomics by enabling us to study the *micro-macro relation*. This relation is not just a linear scaling-up, but can have a complex 'chemical' effect, known as the *emergent property*. Consequently, economics becomes a part of the *Sciences of Emergence*. Second, ACE is able to demonstrate a lively *co-evolution* process, which provides a new platform for testing economic theories. Moreover, what comes with the co-evolution process is a *novelty-generation* process. The latter is, in particular, the weakest area of neoclassical economics.

Micro-Macro Relation and Emergent Properties

Agent-based modeling provides us with a rich opportunity to study the so-called *'micro-macro relation'*, which is beyond the feasibility of the neoclassical economics that consists of only a few representative agents. The first type of micro-macro study involves laying the *foundation* for the aggregate behavior upon the *agents' interacting adaptive schemes*. A series of efforts were made by [17] and [70] to attribute, in an analytical way, the appearance of some interesting macroeconomic phenomena, such as fluctuations in foreign exchange rates, the bid-ask spread, hyperinflation and economic take-off, to the adaptive behavior driven by GA. Elements, such as *self-reinforcement* and *critical mass*, upon which the conventional arguments are built, are actually encapsulated into GAs. [53], [56] and [60], on the other hand, showed the significance of the *survival-of-the-fittest principle* to the convergence to Pareto optimality. In their agent-based cobweb model, OLG model of saving and inflation, and coordination games, it was shown that the property of converging to Pareto optimality will break down if *survival pressure* is removed.

The second type of micro-macro study is concerned with the *consistency* between the micro behavior and the macro behavior. A particularly interesting thing is that the micro behavior can sometimes be quite different from the macro behavior. Both the work done by [81] on the cobweb model and [58] and [59] on the asset pricing model showed that the time series of the market price (an aggregate variable) followed a simple stochastic process. However, there is no simple description of the population dynamics of individual behavior. The simple stochastic price behavior was, in effect, generated by a great diversity of agents whose behavior was constantly changing. [58] proposed a measure for the *complexity* of an agent's behavior and a measure of the *diversity* of an agent's complexity, and it was found that both measures can vary quite widely, regardless of the simple aggregate price behavior.

In addition, using the micro-structure data, [49], [58], [59], and [62] initiated an approach to study what is called the *emergent property*. By that definition, they found that a series of aggregate properties, such as the efficient market hypothesis, the rational expectations hypothesis, the price-volume relation and the sunspot effect, which were proved by rigorous econometric tests, were generated by a majority of agents who did not believe in these properties. Once again, our understanding of the micro behavior does not lead to a consistent prediction of the macro behavior. The latter is simply not just the linear scaling-up of the former. Conventional economics tends to defend the policy issues concerned with the individual's welfare (for instance the national annuity program), based on macroeconometric tests such as the permanent income hypothesis. Agent-based macroeconomics may invalidate this approach due to emergent properties.

Co-Evolution

Briefly, co-evolution means that everything depends on everything else. The performance of one strategy depends on the composition of the strategies with which it interacts, and the fundamental push for agents to adapt arises because other agents are adapting as well. This idea is by no means new to economists. Actually, it is the main subject of evolutionary game theory. However, what has not been shown explicitly in the evolutionary game theory or mainstream economics is that *it is the force of co-evolution which generates novelties*. We shall say a few words concerning their relation here, but more on novelty in the next Section.

Novelties-generation, from its general characteristics to its formation process, is little known in mainstream economics. For example, there is no formal (mathematical) description of how the MS-DOS system eventually led to the MS-Windows system. Neither is there an abstract description showing how commodities $A_1, A_2, \ldots, A_n$ in the early days lead to commodities $B_1, B_2, \ldots, B_m$ at a later stage, or how a population of behavior years ago leads to a different population of behavior at present. Quite ironically, the vision of the 'Father of Neoclassical Economics', Alfred Marshall, namely, "Economics, like biology, deals with a matter, of which the inner nature and constitution, as well as outer form, are constantly changing," was virtually not carried out at all by his offspring (neoclassical economists) [118].

ACE attempts to recast economics along biological and evolutionary lines. Within the co-evolutionary framework, the system which an agent faces is essentially *open* and *incomplete*. The optimal kinds of behavior or strategies which interest most economists may not necessarily exist in this system. In his agent-based cobweb model, [81] used the *survival distribution function* of firms to show *waves of evolutionary activity*. In each wave, one witnesses the sudden collapse of a strongly dominating strategy, the 'optimal' strategy. Very typically, the co-evolution demonstrated in the agent-based model is not a peaceful state of co-existence, but is an incessant struggle for survival where no strategy can be safe from being replaced in the near future. Novel strategies are spontaneously developed and old 'optimal' strategies are continually replaced.

This feature casts doubt on the 'optimal' economic behavior which is not derived from the agent-based co-evolutionary context. In this way, Chen and Huang's agent-based model of investment lent support to the non-optimality of the capital asset pricing model (CAPM) [45]. The optimality of the CAPM was originally derived from a general equilibrium setting. However, they simulated an agent-based multi-asset market, and showed that, in most of their simulations, the fund managers who followed the CAPM did not survive when investors with the constant relative risk aversion presented.

In [45], the CAPM traders and many different types of traders were all introduced to the market right at the beginning (at the initialization stage). They were competing with other agents whose portfolio strategies were evolving over time and which were characterized by GA. The annihilation of the CAPM traders was the result of this setting. This kind of test is referred to as the *formula-agent* approach. Formula agents are agents whose behavior or decision rules are inspired by economic theory. Based on this approach, the economic behavior predicted by economic theory is tested by directly adding formula agents to the initial population. Doing so may be biased because the resultant co-evolution process may be determined by these initial *hints*, a common phenomenon known as *path dependence*.[25] Therefore, the formula-agent approach is relatively *weak* as opposed to an alternative approach to

[25] Path dependence is ubiquitous in ABM. For example, in Dawid's agent-based model of double auctions, the distribution of competitive prices is sensitively dependent on the distribution of initial bids and asks [69], [70].

the co-evolution test, and Lensberg's agent-based model of investment is an illustration of this alternative [112].

Lensberg's model tested *Bayesian rational investment behavior*. However, unlike [45], [112] did not initialize the market with any *Bayesian rational investor*. In other words, all agents' investment rules were generated from scratch (by GP). It was then shown that, in later periods of evolution, what dominated the populations (the surviving firms) were the behavioral rules as if they were expected utility maximizers with Bayesian learning rules. Therefore, the Bayesian rational investment rule was validated as a behavior emerging from the bottom.

However, not all cases have lent support to what economic theory predicts. [110]'s version of the SFI (Santa Fe Institute) artificial stock market is a case in point. *Stationarity* associated with the *asymptotic theory* plays an important role in current developments in econometrics. In the mainstream rational-expectations econometrics, agents are assumed to be able to learn from this stationary environment by using the so-called *Kolmogorov-Wiener filter*. The use of this filter can make sense only if agents believe that the entire time series is stationary, and never doubt that things may have changed in the recent past. Agents with this belief are called *'long-horizon agents'* in LeBaron's ABM. In a similar way to [112], LeBaron questioned whether these long-horizon agents can eventually emerge from the evolution of a population of short-horizon agents, given that the true dividends-generation process is indeed stationary [110]. Interestingly, he found that while long-horizon agents are able to replicate usual efficient market results, evolving a population of short-horizon agents into long-horizon agents is *difficult*. This study, therefore, presents a typical coordination failure problem frequently addressed in macroeconomics.

Within this co-evolution test framework, the maximizing-expected-utility (MEU) behavior of investors, known as the *principle of maximizing certainty equivalence*, was also rejected by [34], [113], and [142]. In their agent-based models of *investment under uncertainty*, they all came up with the same conclusion: *those who survive were not the most efficient in a normative sense* – in other words, the MEU agents were not able to survive. Hence, in a sense, the equivalence between *efficiency* and *survival* broke down. What happened instead was that the surviving investors either took too much risk [113] or were too cautious [34, 142].

Novelties

As mentioned earlier, in ACE, what may come with a co-evolutionary process is a novelties-generation process. This feature is similar to Hayek's evolutionary concept of '*competition as a discovery procedure*.' The neoclassical economic models are completely silent on the novelties-generation process, from their general characteristics to their formation process. One basically cannot anticipate anything unanticipated from the neoclassical model. All types of economic behavior are determined exogenously and can only be renewed manually by the model designers in a top-down manner. This makes it very hard for neoclassical economics to give a constructive notion of preferences, commodities, technology, human capital, and organization, concepts that are fundamentally related to the *theory of economic change*.

Back in the late 1980s, Holland and Arthur had already sketched a research idea, known as '*growing artificial economy*', which was basically to simulate the evolution of an economy from its primitive state to the advanced state. This big plan, however, was never carried out. Instead, what was actually implemented was found in Epstein and Axtell's famous book, '*growing artificial societies*.' In a model of cellular automata, they evolved many interesting kinds of economic and social behavior, including trade, migration, disease, distribution of wealth, social networks, sexual reproduction, cultural processes, and combat. In addition to this major piece of work, [132] studied how money as a medium of exchange can emerge from a bartering economy, and [17] also simulated the appearance of an economic take-off (the industrial revolution).

Despite these studies, one has to say that the novelties-generation process has not been well exploited given the current state of ABM. There should be more left for the researchers to do. In their research project, the *functional-modularity foundation of economics*, [41, 42] proposed an agent-based model of preference changes and technology formation to *grow* both technology and preferences. In their model, consumers' current preferences will determine the direction of technology advancement. However, the technology developed will in turn evolve the preferences as well. GP is applied here to give size-free and shape-free representation of technology and preferences.

The use of genetic programming in economics provides economists a great opportunity to rethink some hundred-year-old ideas. In particular, it enables economists to implement the ideas of economic evolution or progress by incorporating and hence acknowledging the importance of modularity. Simulating economic evolution with functional modularity is not restricted to technology or product innovation. More challenging tasks are its application to labor markets and organizations. While the idea that labor as capital (known as 'human capital'), has been studied for almost 40 years, the process of accumulating human capital – and hence the role of education, as well as on-job training – is yet to be established.

5.3 Future Directions

Before ending this Section, we would like point out some directions for further research so as to see more opportunities and challenges opening for applications of computational intelligence tools.

Experimental Economics and Behavioral Economics

It becomes gradually clear that agent-based computational economics should be able to interact with experimental and behavioral economics in a more integrated framework. A series of papers published recently motivated the need for an integrated framework and sketched how this work can be done. First, the behavioral approach and the agent-based approach can collaboratively work together in a bi-directional manner. On the one hand, experimental and behavioral approaches can help answer some modeling issues related to agent engineering, while, on the other hand, agent-based computational finance can help test the robustness or the generality of some behavioral rules observed from psychological laboratory experiments.

[43] serves as an example of the first direction. While the essence of agent-based computing is agents, not so much has been said as to how to model or program these agents. Disputes still prevail on the issue like the simple/naive agents versus the sophisticated/smart agents.[26] A proposed solution to this problem is to work with real human behavior, in particular, when the respective fields or an experimental study are available. For example, there are already some empirical observations regarding gamblers' behavior; hence, one may get some ideas on how a gambling agent should be programmed in light of the empirical evidence. Chen and Chie's work on the agent-based modeling of lottery markets serves as a demonstration of this idea [43].

[48] serves as an example of the other direction. Psychologists have been long distinct from economists in the rational assumption of human behavior. The gap between the two has, however, been narrowed in recent years, thanks to a series of celebrated works by Tverskey, Kahneman, and their followers. Findings based on a series of psychological experiments concerning decision-making under risk and uncertainty are now applied to address a number of financial anomalies, which fosters the growing field currently known as *behavioral finance.*

Chen and Liao, however, questioned the legitimacy of financial models directly built upon psychological experiments [48]. Their main concern is that psychological experiments which lend support to various cognitive biases seem to focus only on independent individual behavior in a rather static environment. This setting is, therefore, distant from financial markets, where agents are able to learn and adapt in an interactively dynamic environment. As a result, various cognitive biases observed from the psychological experiments may be corrected via learning and may not be exogenously fixed as in most behavioral financial models. [48] proposed an alternative: instead of exogenously imposing a specific kind of behavioral bias (for example overconfidence or conservatism) on the agents, we can canvass the emergence and/or

[26] See [50] for an in-depth discussion of this issue.

the survivorship of this behavioral bias in the highly dynamic and complex environment through computer simulations.

Second, when software agents are commonly used to replace human agents in making decisions and taking action in an era of electronic commerce, human agents and software agents can quite often be placed in a common arena and their interaction becomes more intense than ever. Questions pertaining to the consequences of this interaction, therefore, become crucial. [87] pioneered such a research direction, and raised two fundamental issues which define this line of research. First, will artificial agents in markets influence human behavior? Second, will the interaction between human and artificial agents have a positive or negative effect on the market's efficiency? They designed a continuous double auction market in the style of the Iowa electronic market, and introduced software agents with a passive arbitrage seeking strategy to the market experiment with human agents. Whether or not the human agents are well informed of the presence of the software agents can have significant impacts upon market efficiency (in the form of price deviations from the fundamental price). They found that if human agents are well informed, then the presence of software agents triggers more efficient market prices when compared to the baseline treatment without software agents. Otherwise, the introduction of software agents results in lower market efficiency.[27]

Agent-Based Econometric Modeling

We now have seen some progress regarding how agent-based models can be built upon laboratory experiments with human subjects, field studies, and social work, but not directly with the data themselves. This is concerned with agent-based econometric models. The complexity of the agent-based models makes their empirical estimation a daunting task, if not an impossible one. Therefore, few attempts have been made to conduct an econometric analysis of an agent-based model. However, recently, we have started to see some progress in the estimation of some relatively simple agent-based models; [122] was one of the pioneering efforts.

[122] can be regarded as an outcome of the new research trend that embeds conventional discrete choice models, also known as the qualitative response models, in a social network, and examines the impact of the social interaction upon individuals' discrete choices. Other similar works can be found in [28] and [74]. With moderate degrees of simplifying assumptions on individuals' decision models as well as interaction mechanisms, this network-based agent-based model can be parameterized and estimated as an econometric model. This is basically what was done in [122], which estimated the interaction mechanism among young people in relation to smoking behavior by using the

[27] These two issues have been further pursued in the recent development of the `U-Mart` platform [103, 133, 145].

result of [8].[28] Its empirical results strongly support the presence of positive peer effects in smoking behavior among young people.

Certainly, not all agent-based econometric models are network-based. There are a series of agent-based financial econometric models which do not explicitly refer to a network or a graph [4, 5, 155].

Agent-Based Social Networks

Having noticed that agent-based econometric models were first successfully developed in the area of the network-based discrete choice models, we noticed that the social network plays an increasingly important role in ACE models. In fact, the network should be an essential ingredient of agent-based models, while most agent-based simulation models do not explicitly include this element.

Network externalities may be viewed as one of the best places to see the use of agent-based social networks. The celebrated work [97] was demonstrated in an agent-based manner by [148], who built an agent-based model and evaluated it by verifying the simulation results with conventional Beta and VHS systems. [83] enhances our understanding of the significance of network effects by creating agent-based computational simulations of such markets. Insights into the dominance of the inferior technologies are further explored within a model called 'Standard-Scape'.

6 Concluding Remarks

Unlike most tutorials on the economic and financial applications of computational intelligence, this Chapter is not about how CI tools are applied to economics and finance as merely an optimization numerical tool. Instead, we have a broader scope, namely to use CI tools to build economic agents with some reasonable and realistic degree of autonomy, and then study the emergent phenomena resulting from a society of these autonomous agents. In this sense, CI is introduced to economics as an algorithmic foundation of autonomous agents. We review two major algorithmic foundations, namely, neural networks and evolutionary computation. While the review is not exhaustive, the essential idea of using other tools to build autonomous agents and hence evolve the economy is largely the same. A well-motivated reader should be able to see room for other alternative algorithmic foundations, such

[28] [8] can be read as one of the earliest available econometric results of agent-based models. Given the profile of individual attributes and the social interaction mechanism, [8] provides an analytical solution for the equilibrium distribution of the collection of individuals' behavior. Hence, it is possible to describe the macro equilibrium from the micro level.

as fuzzy logic, decision trees, Bayesian networks, and reinforcement learning. The general question left for further study is how these different CI tools, under what specific environment, can successfully characterize human decision-making process.

In the second part of this Chapter, we reviewed the early applications of CI to agent-based computational economics. We saw how CI can help relax the stringent assumptions frequently used in old-fashion economics, and bring back some missing processes due to the learning or bounded rational behavior of agents. While making economic predictions using ACE models is still difficult, the use of CI certainly enhance some of our flexibility to simulate possible futures. In this review, we have witnessed how CI can help build more realistic ACE models such that they can be useful in policy design.

The nature of economics is change and evolution, and what makes it change and evolve is humans. CI provides alternative hypotheses or modeling of the microscopic details of human behavior. So long as these details are not trivial, CI can help economists to establish quality models as illustrated in the many works reviewed in this Chapter.

Acknowledgements

The author is grateful to one of the anonymous referees for their helpful suggestions. The author is also grateful to Professor John Fulcher for his painstaking efforts made in editing the Chapter. Research support in the form of NSC grant No. NSC. 95-2415-H-004-002-MY3 is gratefully acknowledged.

References

1. Adcock A, Thangavel1 A, Whitfield-Gabrieli S, Knutson B, Gabrieli J (2006) Reward-motivated learning: Mesolimbic activation precedes memory formation. *Neuron,* 50(3): 507–517.
2. Aha D (1997) *Lazy Learning.* Kluwer, Boston, MA.
3. Aha D, Kibler D, Marc K (1991) Instance-based learning algorithms. *Machine Learning,* 6(1): 37–66.
4. Alfarano S, Lux T, Wagner F (2005) Estimation of agent-based models: the case of an asymmetric herding model. *Computational Economics*, 26(1): 19–49.
5. Alfarano S, Lux T, Wagner F (2007) Empirical validation of stochastic models of interacting agents: a 'maximally skewed' noise trader model. *European Physics J. B,* 55: 183–187.
6. Allen F, Karjalainen R (1999) Using genetic algorithms to find technical trading rules. *J. Financial Economics,* 51(2): 245–71.
7. Alvarez-Diaz M, Alvarez A (2005) Genetic multi-model composite forecast for nonlinear prediction of exchange rates. *Empirical Economics*, 30: 643–663.
8. Amemiya T (1975), Qualitative response models. *Annals Economics and Social Management*, 4: 363–372.

9. Andrew M, Prager R (1994) Genetic programming for the acquisition of double auction market strategies. In: Kinnear K Jr. (ed.), *Advances in Genetic Programming*. MIT Press, Cambridge, MA: 355–368.
10. Arifovic J (1994) Genetic algorithms learning and the cobweb model. *J. Economic Dynamics and Control,* 18(1): 3–28.
11. Arifovic J (1995) Genetic algorithms and inflationary economies. *J. Monetary Economics*, 36(1): 219–43.
12. Arifovic J (1996) The behavior of the exchange rate in the genetic algorithm and experimental economies. *J. Political Economy,* 104(3): 510–541.
13. Arifovic J (2001) Evolutionary dynamics of currency substitution. *J. Economic Dynamics and Control,* 25: 395–417.
14. Arifovic J (2002) Exchange rate volatility in the artificial foreign exchange market. In: Chen S-H (ed.) *Evolutionary Computation in Economics and Finance.* Physica-Verlag, Berlin: 125–136.
15. Arifovic J, Eaton B (1995) Coordination via genetic learning. *Computational Economics,* 8(3): 181–203.
16. Arifovic J, Gencay R (2000) Statistical properties of genetic learning in a model of exchange rate. *J. Economic Dynamics and Control,* 24: 981–1005.
17. Arifovic J, Bullard J, Duffy J (1997) The transition from stagnation to growth: an adaptive learning approach. *J. Economic Growth,* 2(2): 185–209.
18. Armano G, Murru A, Marchesi M (2002) NXCS – A hybrid approach to stock indexes forecasting. In: Chen, S-H (ed.) *Genetic Algorithms and Genetic Programming in Computational Finance.* Kluwer, Boston, MA.
19. Arthur B (1992) On learning and adaptation in the economy. *SFI Economics Research Program,* 92-07-038.
20. Arthur W, Holland J, LeBaron B, Palmer R, Tayler P (1997) Asset pricing under endogenous expectations in an artificial stock market. In: Arthur W, Durlauf S, Lane D (eds.) *The Economy as an Evolving Complex System II.* Addison-Wesley, Reading, MA: 15–44.
21. Axelrod R (1997) Advancing the art of simulation in the social sciences. In: Conte R, Hegselmann R, Terna P (eds.) *Simulating Social Phenomena.* Springer-Verlag, Berlin: 21–40.
22. Azariadis C, Guesnerie R (1986) Sunspots and cycle. *Review Economic Studies LIII*: 725–737.
23. Azoff M (1994) *Neural Network Time Series: Forecasting of Financial Markets.* Wiley, New York, NY.
24. Baestaens D, Van Den Bergh W, Wood D (1994) *Neural Network Solutions for Trading in Financial Markets*. Pitman, London, UK.
25. Bauer R. Jr (1994) *Genetic Algorithms and Investment Strategies.* Wiley, New York, NY.
26. Bell R, Beare S (2002) Emulating trade in emissions permits: An application of genetic algorithms. In: Chen S-H (ed.) *Evolutionary Computation in Economics and Finance*. Physica-Verlag, Heidelberg, Germany: 161–175.
27. Birchenhall C, Lin J-S (2002) Learning and convergence to Pareto optimality. In: Chen S-H. (ed.) *Genetic Algorithms and Genetic Programming in Computational Finance.* Kluwer, Boston, MA: 419–440.
28. Bonabeau E (2003) Econometrics of agent-based models. *Proc. 2nd Lake Arrowhead Conf. Human Complex Systems (Keynote Speech),* 19–22 March, Lake Arrowhead, CA.

29. Bower J, Bunn D (2001) Experimental analysis of the efficiency of uniform-price versus discriminatory auctions in the England and Wales electricity market. *J. Economic Dynamics and Control,* 25: 561–592.
30. Boyle S, Guerin S, Kunkle D (2005) An application of multi-agent simulation to policy appraisal in the criminal justice system. In: Chen S-H, Jain LC, Tai C-C (eds.) *Computational Economics: A Perspective from Computational Intelligence.* Idea Group, Hershey, PA: 228–234.
31. Bullard J, Duffy J (1998) A model of learning and emulation with artificial adaptive agents. *J. Economic Dynamics and Control,* 22: 179–207.
32. Bullard J, Duffy J (1998) Learning and the stability of cycles. *Macroeconomic Dynamics,* 2(1): 22–48.
33. Bullard J, Duffy J (1999) Using genetic algorithms to model the evolution of heterogeneous beliefs. *Computational Economics,* 13(1): 41–60.
34. Cacho O, Simmons P (1999) A genetic algorithm approach to farm investment. *Australian J. Agricultural and Resource Economics,* 43(3): 305–322.
35. Chan N, LeBaron B, Lo A, Poggio T (1999). Agent-based models of financial markets: a comparison with experimental markets. *Unpublished Working Paper,* MIT Artificial Markets Project, MIT, MA.
36. Chen J, Xu D (1998) An economic forecasting system based on recurrent neural networks. In: *Proc. IEEE Intl. Conf. Systems, Man, and Cybernetics,* 14–18 October, San Diego, CA. IEEE Press, New York, NY, 2: 1762–1767.
37. Chen S-H (1997) On the artificial life of the general economic system (I): the role of selection pressure. In: Hara F, Yoshida K (eds.) *Proc. Intl. Symp. System Life,* 21–22 July, Tokyo, Japan: 233–240.
38. Chen S-H (2000) Toward an agent-based computational modeling of bargaining strategies in double auction markets with genetic programming. In: Leung K-S, Chan L-W, Meng H (eds.) *Intelligent Data Engineering and Automated Learning – IDEAL 2000: Data Mining, Financial Engineering, and Intelligent Agents,* Lecture Notes in Computer Science 1983. Springer-Verlag, Berlin: 517–531.
39. Chen S-H (ed.) (2002) *Evolutionary Computation in Economics and Finance.* Physica-Verlag, Berlin.
40. Chen S-H (ed.) (2002) *Genetic Algorithms and Genetic Programming in Computational Finance.* Kluwer, Boston, MA.
41. Chen S-H, Chie B-T (2004) Agent-based economic modeling of the evolution of technology: the relevance of functional modularity and genetic programming. *Intl. J. Modern Physics B,* 18(17–19): 2376–2386.
42. Chen S-H, Chie B-T (2005) A functional modularity approach to agent-based modeling of the evolution of technology. In: Namatame A, Kaizouji T, Aruka Y (eds.) *The Complex Networks of Economic Interactions: Essays in Agent-Based Economics and Econophysics,* Lecture Notes in Economics and Mathematical Systems 567, Springer-Verlag, 165–178.
43. Chen S-H, Chie B-T (2007) Lottery markets design, micro-structure, and macro-behavior: An ACE approach. *J. Economic Behavior and Organization,* (in press).
44. Chen S-H, He H (2003) Searching financial patterns with self-organizing maps. In: Chen S-H, Wang P (eds.) *Computational Intelligence in Economics and Finance.* Springer-Verlag, Berlin: 203–216.

45. Chen S-H, Huang Y-C (2007) Risk preference, forecasting accuracy and survival dynamics: simulations based on a multi-asset agent-based artificial stock market. *J. Economic Behavior and Organization.* (in press).
46. Chen S-H, Kuo T-W (1999) Towards an agent-based foundation of financial econometrics: an approach based on genetic-programming artificial markets. In: Banzhaf W, Daida J, Eiben A, Garzon M, Honavar V, Jakiela M, Smith R (eds.) *Proc. Genetic and Evolutionary Computation Conf.*, Morgan Kaufmann, San Mateo, CA, 2: 966–973.
47. Chen S-H, Liao C-C (2002) Price discovery in agent-based computational modeling of artificial stock markets. In: Chen S-H (ed.) *Genetic Algorithms and Genetic Programming in Computational Finance.* Kluwer, Boston, MA: 333–354.
48. Chen, S-H, Liao C-C (2004) Behavior finance and agent-based computational finance: toward an integrating framework. *J. Management and Economics,* 8.
49. Chen S-H, Liao C-C (2005) Agent-based computational modeling of the stock price-volume relation. *Information Sciences,* 170: 75–100.
50. Chen S-H, Tai C-C (2006) On the selection of adaptive algorithms in ABM: a computational-equivalence approach. *Computational Economics,* 28(1): 51–69.
51. Chen S-H, Wang P (eds.) (2003) *Computational Intelligence in Economics and Finance.* Springer-Verlag, Berlin.
52. Chen S-H, Yeh C.-H (1996) Genetic programming and the efficient market hypothesis. In: Koza J, Goldberg D, Fogel D, Riolo R (eds.) *Genetic programming 1996: Proc. 1st Annual Conf.* MIT Press, Cambridge, MA: 45–53.
53. Chen S-H, Yeh C-H (1996) Genetic programming learning and the cobweb model. In: Angeline P (ed.) *Advances in Genetic Programming 2.* MIT Press, Cambridge, MA: 443–466.
54. Chen S-H, Yeh C-H (1997) Toward a computable approach to the efficient market hypothesis: an application of genetic programming. *J. Economic Dynamics and Control,* 21: 1043–1063.
55. Chen S-H, Yeh C-H (1997) Modeling speculators with genetic programming. In: Angeline P, Reynolds R, McDonnell J, Eberhart R (eds.) *Evolutionary Programming VI,* Lecture Notes in Computer Science 1213. Springer-Verlag, Berlin: 137–147.
56. Chen S-H, Yeh C-H (1999) Modeling the expectations of inflation in the OLG model with genetic programming. *Soft Computing,* 3(2): 53–62.
57. Chen S-H, Yeh C-H (2000) Simulating economic transition processes by genetic programming. *Annals Operation Research,* 97: 265–286.
58. Chen S-H, Yeh C-H (2001) Evolving traders and the business school with genetic programming: a new architecture of the agent-based artificial stock market. *J. Economic Dynamics and Control,* 25: 363–393.
59. Chen S-H, Yeh C-H (2002) On the emergent properties of artificial stock markets: the efficient market hypothesis and the rational expectations hypothesis. *J. Economic Behavior and Organization,* 49(2): 217–239.
60. Chen S-H, Duffy J, Yeh C-H (2001) Equilibrium selection via adaptation: using genetic programming to model learning in a co-ordination game. *Electronic J. Evolutionary Modeling and Economic Dynamics.* 1: 1002.
61. Chen, S-H, Lee W-C, Yeh C-H (1999) Hedging derivative securities with genetic programming. *Intl. J. Intelligent Systems in Accounting, Finance and Management,* 8(4): 237–251.

62. Chen S-H, Liao C-C, Chou P-J (2007) On the plausibility of sunspot equilibria: simulations based on agent-based artificial stock markets. *J. Economic Interaction and Coordination,* (in press).
63. Chen S-H, Yeh C-H, Liao C-C (2002) On AIE-ASM: Software to simulate artificial stock markets with genetic programming. In: Chen S-H (ed.) *Evolutionary Computation in Economics and Finance.* Physica-Verlag, Heidelberg, Germany: 107–122.
64. Chen T, Chen H (1995) Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans. Neural Networks,* 6: 911–917.
65. Chidambaran N, Lee C, Trigueros J (2000) Option pricing via genetic programming. In: Abu–Mostafa Y, LeBaron B, Lo A, Weigend A (eds.) *Computational Finance – Proc. 6th Intl. Conf.* MIT Press, Cambridge, MA: 583–598.
66. Cliff D, Bruten J (1997) Minimal-intelligence agents for bargaining behaviors in market-based environments. *Technical Report 97-91,* Hewlett-Packet Lab.
67. Cristianini N, Shawe-Taylor J (2000) *An Introduction to Support Vector Machines – and Other Kernel-Based Learning Methods.* Cambridge University Press, UK.
68. Das N (2003) Hedge fund classification using k-means method. *EconPapers,* 204: 284.
69. Dawid H (1996) Learning of cycles and sunspot equilibria by genetic algorithms. *J. Evolutionary Economics,* 6(4): 361–373.
70. Dawid H (1999) On the convergence of genetic learning in a double auction market. *J. Economic Dynamics and Control,* 23: 1544–1567.
71. Deboeck G, Kohonen T (1998) *Visual Explorations in Finance with Self-Organizing Maps.* Springer-Verlag, Berlin.
72. Duffy J (2001) Learning to speculate: experiments with artificial and real agents. *J. Economic Dynamics and Control,* 25: 295–319.
73. Duffy J (2006) Agent-based models and human-subject experiments. In: Tesfatsion L, Judd K (eds.) *Handbook of Computational Economics 2.* North Holland, Amsterdam, The Netherlands.
74. Dugundji E, Gulyas L (2003) Empirical estimation and multi-agent-based simulation of a discrete choice model with network interaction effects. In: Macal C, North M, Sallach D (eds.) *Proc. Agent 2003 Conf. on Challenges in Social Simulation,* 2–4 October, University of Chicago. Argonne National Laboratory, Chichago, IL: 437–453.
75. Eberhart R, Simpson P, Dobbins R (1996) *Computational Intelligence PC Tools.* Academic Press, Boston, MA.
76. Elman J (1990) Finding structure in time. *Cognitive Science,* 14: 179–211.
77. Episcopos A, Davis J (1996). Predicting returns on Canadian exchange rates with artificial neural networks and EGARCHM-M model. *Neural Computing and Applications,* 4:168–174.
78. Fernández-Rodríguez F, Sosvilla-Rivero S, Andrada-Félix J (2003) Nearest-neighbour predictions in foreign exchange markets. In: Chen S-H, Wang P (eds.) *Computational Intelligence in Economics and Finance.* Springer-Verlag, Berlin: 297–325.
79. Fogel D (1995) *Evolutionary Computation – Toward a New Philosophy of Machine Intelligence.* IEEE Press, Piscatawy NJ.
80. Fogel L (1964) On the Organization of Intellect. *PhD Thesis,* University of California at Los Angeles, CA.

81. Franke R (1998) Coevolution and stable adjustments in the cobweb model. *J. Evolutionary Economics,* 8(4): 383–406.
82. Fogel L, Owens A, Walsh M (1966) *Artificial Intelligence through Simulated Evolution.* Wiley, New York, NY.
83. Frels J, Heisler D, Reggia J (2003) Standard-scape: An agent-based model of adoption with incomplete information and network externalities. In: Chen K, Chen S-H, Cheng H-D, Chiu D, Das S, Duro R, Jiang Z, Kasabov N, Kerre E, Leong H-V, Li Q, Lu M, Romay MG, Ventura D, Wang P-P, Wu J (eds.) *Proc. 7th Joint Conf. Information Sciences,* 26–30 September, Cary, NC: 1219–1222.
84. Gately E (1996) *Neural Networks for Financial Forecasting.* Wiley, New York, NY.
85. Gode D, Sunder S (1993) Allocative efficiency of markets with zero-intelligence traders: market as a partial substitute for indiviudal rationality. *J. Political Economy,* 101(1): 119–137.
86. Grandmont J-M (1985) On endogeneous competitive business cycles. *Econometrica,* 53: 995–1045.
87. Grossklags J, Schmidt C (2006) Software agents and market (in)efficiency: a human trader experiment. *IEEE Trans. Systems, Man, and Cybernetics, Part C,* 36(1): 56–67.
88. Grossman S (1976) On the efficiency of competitive stock markets where traders have diverse information. *J. Finance,* 31: 573–585.
89. Grossman S, Stiglitz J (1980) On the impossibility of informationally efficient markets. *American Economic Review,* 70: 393–408.
90. Hann T, Steurer E (1996). Much ado about nothing? Exchange rate forecasting: Neural networks vs. linear models using monthly and weekly data. *Neurocomputing,* 10: 323–339.
91. Holland J, Holyoak K, Nisbett R (1986) *Induction: Processes of inference, learning and discovery (computational models of cognition and perception).* MIT Press, Cambridge, MA.
92. Holland J, Miller J (1991) Artificial adaptive agents in economic theory. *American Economic Review,* 81(2): 365–370.
93. Hollans H, Munneke H. (2003) Housing markets and house price appreciation: An Interacity Analysis. *Working paper.* University of Georgia.
94. Izumi K, Ueda K (1999) Analysis of dealers' processing financial news based on an artificial market approach. *J. Computational Intelligence in Finance,* 7: 23–33.
95. Jordan M (1986) Serial order: A parallel distributed processing approach. *Technical Report 8604.* Institute for Cognitive Science, University of California.
96. Kareken J, Wallace N (1981) On the indeterminacy of equilibrium exchange rate. *Quarterly J. Economics,* 96: 207–222.
97. Katz M, Shapiro C (1985) Network externalities, competition, and compatibility. *American Economic Review,* 75: 424–440.
98. Keber C (1999) Genetically derived approximations for determining the implied volatility. *OR Spektrum,* 21: 205–238.
99. Keber C (2000) Option valuation with the genetic programming approach. In: Abu-Mostafa Y, LeBaron B, Lo A, Weigend A (eds.) *Computational Finance – Proc. 6th Intl. Conf.* MIT Press, Cambridge MA.
100. Kiyotaki N, Wright R (1989) On money as a medium of exchange. *J. Political Economy,* 97: 927–954.

101. Kohonen T (1982) Self-organized foundation of topologically correct feature maps. *Biological Cybernetics,* 43: 59–69.
102. Kohonen T (1995) *Self-Organizing Maps.* Springer-Verlag, Berlin.
103. Koyama Y, Sato H, Matusi H, Nakajima Y (2005) Report on UMIE 2004 and summary of U-Mart experiments based on the classification of submitted machine agents. In: Terano T, Kita H, Kaneda T, Arai K, Deghchi H (eds.) *Agent-Based Simulation: From Modeling Methodologies to Real-World Applications,* (Springer Series on Agent-Based Social Systems 1), Springer-Verlag, Tokyo: 158–166.
104. Koza J (1992) *Genetic programming: on the programming of computers by means of natural selection.* MIT Press, Cambridge, MA.
105. Koza J (1992) A Genetic approach to econometric modeling. In: Bourgine P, Walliser B (eds.) *Economics and Cognitive Science.* Pergamon Press, Oxford, UK: 57–75.
106. Kramer M (1990) Nonlinear principal analysis using autoassociative neural networks. *AIChE J.,* 37(2): 233–243.
107. Krugman P (1996) *The Self-Organizing Economy.* Blackwell, Cambridge, MA.
108. Kuan C-M, Liu T (1995) Forecasting exchange rates using feedforward and recurrent neural networks. *J. Applied Econometrics,* 10: 347–364.
109. LeBaron B (1999) Building financial markets with artificial agents: Desired goals and present techniques. In: Karakoulas G (ed.) *Computational Markets.* MIT Press, Cambridge, MA.
110. LeBaron B (2001) Evolution and time horizons in an agent based stock market. *Macroeconomic Dynamics,* 5: 225–254.
111. LeBaron B, Arthur W, Palmer R (1999) Time series properties of an artificial stock market. *J. Economic Dynamics and Control,* 23: 1487–1516.
112. Lensberg T (1999) Investment behavior under Knightian uncertainty – an evolutionary approach. *J. Economic Dynamics and Control,* 23: 1587–1604.
113. Lettau M (1997) Explaining the facts with adaptive agents: the case of mutual fund flows. *J. Economic Dynamics and Control,* 21(7): 1117–1147.
114. Lucas R (1986) Adaptive behavior and economic theory. In: Hogarth R, Reder M (eds.) *Rational choice: The contrast between economics and psychology.* University of Chicago Press, IL: 217–242.
115. MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: LeCam LM, Neyman N (eds.) *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability.* University of California Press, Berkeley, CA, 1: 281–297.
116. Mandic D, Chambers J (2001). *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures, and Stability.* Wiley, New York, NY.
117. Marimon R, McGrattan E, Sargent T (1990) Money as Medium of Exchange in an Economy with Artificially Intelligent Agents. *J. Economic Dynamics and Control,* 14: 329–373.
118. Marshall A (1961) *Principles of Economics (9th (variorum) ed., with annotations by CW Guillebaud).* Macmillan, London, UK.
119. McNelis P (2005). *Neural Networks in Finance: Gaining Predictive Edge in the Market.* Elesvier, Burlington, MA.
120. Midgley G (2005) Systemic intervention for community involvement in complex policy. In: *Proc. 1st Intl. Workshop Complexity and Policy Analysis,* 22–24 June, Cork, Ireland.

121. Muth J (1961) Rational expectations and the theory of price movements. *Econometrics,* 29: 315–335.
122. Nakajima R (2003) Measuring peer effects in youth smoking behavior. In: Chen K, Chen S-H, Cheng H-D, Chiu D, Das S, Duro R, Jiang Z, Kasabov N, Kerre E, Leong H-V, Li Q, Lu M, Romay MG, Ventura D, Wang P-P, Wu J (eds.) *Proc. 7th Joint Conf. Information Sciences,* 26-30 September, Cary, NC: 1206–1210.
123. Neely C, Weller P, Ditmar R (1997) Is technical analysis in the foreign exchange market profitable? A genetic programming approach. *J. Financial and Quantitative Analysis,* 32(4): 405–427.
124. North M (2003) Applying computational Intelligence to economic policy. In: Chen K, Chen S-H, Cheng H-D, Chiu D, Das S, Duro R, Jiang Z, Kasabov N, Kerre E, Leong H-V, Li Q, Lu M, Romay MG, Ventura D, Wang P-P, Wu J (eds.) *Proc. 7th Joint Conf. Information Sciences,* 26-30 September, Cary, NC: 1231–1234.
125. Nicolaisen J, Smith M, Petrov V, Tesfatsion L (2000) Concentration and capacity effects on electricity market power. In: Alzala A (ed.) *Proc. 2000 Congress Evolutionary Computation,* IEEE Society Press, Piscataway, NJ: 1041–1047.
126. Nicolaisen J, Petrov V, Tesfatsion L (2001) Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. *IEEE Trans. Evolutionary Computation,* 5(5): 504–523.
127. Palmer R, Arthur W, Holland J, LeBaron B, and Tayler P (1994). Artificial economic life: a simple model of a stock market. *Physica D,* 75: 264–274.
128. Qian Y (2006) k-means algorithm and its application for clustering companies listed in Zhejiang province. *WIT Trans Information and Communication Technologies,* 37: 35–44.
129. Rechenberg I (1965) Cybernetic Solution Path of an Experimental Problem. *Royal Aircraft Establishment,* Library Translation 1122, August.
130. Refenes A (1995) *Neural Networks in the Capital Markets.* Wiley, New York, NY.
131. Refenes A, Zapranis A (1999) *Principles of Neural Model Identification, Selection and Adequacy: With Applications in Financial Econometrics.* Springer-Verlag, Berlin.
132. Sargent T (1993) *Bounded Rationality in Macroeconomics.* Oxford University Press, UK.
133. Sato H, Kawachi S, Namatame A (2003) The statistical properties of price fluctuations by computer agents in a U-Mart virtual future market simulator. In: Terano T, Dehuchi H, Takadama K (eds.) *Meeting the Challenge of Social Problems via Agent-Based Simulation.* Springer-Verlag, Tokyo: 67–76.
134. Schwefel H (1965) Kybernetische Evolution als Strategies der Experimentellen Forschung in der Strömungstechnik. *Diploma Thesis,* Technical University of Berlin.
135. Schewfel H (1995) *Evolution and Optimum Seeking.* Wiley, New York, NY.
136. Shadbolt J, Taylor J (2002) *Neural Networks and the Financial Markets-Predicting, Combining, and Portfolio Optimisation.* Springer-Verlag, Berlin.
137. Shi S, Xu L, Liu B (1999) Improving the acuracy of nonlinear combined forecasting using neural networks. *Expert Systems with Applications,* 16: 49–54.
138. Simon HA (1997) *Models of Bounded Rationality, Vol. 3.* MIT Press, Cambridge, MA.

139. Smith V, Suchanek G, Williams A (1988) Bubbles, crashes, and endogenous expectations in experimental spot asset markets. *Econometrica,* 56(6): 1119–1152.
140. Suykens J, Vandewalle J (1998) The K.U. Leuven time series prediction competition. In: Suykens J, Vandewalle J (eds.) *Nonlinear Modeling: Advanced Black-Box Techniques.* Kluwer, Boston, MA: 241–253.
141. Szpiro G (1997) Forecasting chaotic time series with genetic algorithms. *Physical Review E,* 55: 2557–2568.
142. Szpiro G (1997) The emergence of risk aversion. *Complexity,* 2: 31–39.
143. Tay N, Linn S (2001) Fuzzy inductive reasoning, expectation formation and the behavior of security prices. *J. Economic Dynamics and Control,* 25: 321–361.
144. Tayler P (1995) Modeling artificial stock markets using genetic algorithms. In: Goonatilake S, Treleaven P (eds.) *Intelligent Systems for Finance and Business.* Wiley, New York, NY: 271–287.
145. Terano T, Shiozawa Y, Deguchi H, Kita H, Matsui H, Sato H, Ono I, Kakajima Y (2003), U-Mart: An artificial market testbed for economics and multiagent systems. In: Terano T, Dehuchi H, Takadama K (eds.) *Meeting the Challenge of Social Problems via Agent-Based Simulation.* Springer-Verlag, Tokyo: 53–66.
146. Tirole J. (1982) On the possibility of speculation under rational expectations. *Econometrica,* 50: 1163–1182.
147. Trippi R, Turban E. (1993) *Neural Networks in Finance and Investing.* Irwin
148. Tsuji M, Kawamura H, Ohuchi A (2003) Measuring the effect of indirect network externality in VCR standardization process. In: Chen K, Chen S-H, Cheng H-D, Chiu D, Das S, Duro R, Jiang Z, Kasabov N, Kerre E, Leong H-V, Li Q, Lu M, Romay MG, Ventura D, Wang P-P, Wu J (eds.) *Proc. 7th Joint Conf. Information Sciences,* 26–30 September, Cary, NC: 1215–1218.
149. Vapnik V (1998) *Statistical Learning Theory.* Wiley, New York, NY.
150. Vapnik V (1998) The support vector method of function estimation. In: Suykens J, Vandewalle J (eds.) *Nonlinear Modeling: Advanced Black-Box Techniques.* Kluwer, Boston, MA: 55–85.
151. Vriend N (2001) On two types of GA-Learning. In: Chen S-H (ed.) *Evolutionary Computation in Economics and Finance.* Physica-Verlag, Heidelberg, Germany: 233–243.
152. Waldrop M (1992) *Complexity: The Emerging Science at the Edge of Order and Chaos.* Simon and Schuster, New York, NY.
153. Wei W-X, Jiang Z-H (1995) Artificial neural network forecasting model for exchange rate and empirical analysis. *Forecasting,* 2: 67–69.
154. Weigend A, Huberman B, Rumelhart D (1992) Predicting sunspots and exchange rates with connectionist networks. In: Casdagli M, Eubank S (eds.) *Nonlinear Modeling and Forecasting.* Addison-Wesley, Reading, MA: 395–432.
155. Westerhoff F, Reitz S (2003) Nonlinearities and cyclical behavior: The role of chartists and fundamentalists. *Studies in Nonlinear Dynamics and Econometrics,* 7(4): Article 3.
156. White H (1988) Economic prediction using neural networks: The case of IBM daily stock returns. In: *Proc. IEEE Intl. Conf. Neural Networks 2*, 24–27 July, San Diego, CA. IEEE Press, New York, NY: 451–458.
157. White H (1992) *Artificial Neural Networks–Approximation Theory and Learning Theory.* Blackwell, Cambridge, MA.

158. Wu B (1995) Model-free forecasting for nonlinear time series (with application to exchange rates). *Computational Statistics and Data Analysis,* 19: 433–459.
159. Wu S, Bhattacharyya S (2005) Minimal intelligence Agents in double auction markets with speculators. In: Chen S-H, Jain LC, Tai C-C (eds.) *Computational Economics: A Perspective from Computationl Intelligence.* Idea Group, Hershey, PA: Chapter IV.
160. Yang J. (2002) The efficiency of an artificial double auction stock market with neural learning agents. In: Chen S-H. (ed.) *Evolutionary Computation in Economics and Finance.* Physica-Verlag, Heidelberg, Germany: 87–107.
161. Yeh C-H, Chen S-H (2001) Toward an integration of social learning and individual learning in agent-based computational stock markets: The approach based on population genetic programming. *J. Management and Economics,* 5.
162. Yeh C-H, Chen S-H (2001) Market diversity and market efficiency: The approach based on genetic programming. *J. Artificial Simulation of Adaptive Behavior,* 1(1): 147–167.
163. Zirilli J (1996) *Financial Prediction Using Neural Networks.* Thomson, London, UK.

Resources

1 Key Books

Arthur W, Holland J, LeBaron B, Palmer R, Tayler P (1997) Asset pricing under endogenous expectations in an artificial stock market. In: Arthur W, Durlauf S, Lane D (eds.) *The Economy as an Evolving Complex System II.* Addison-Wesley, Reading, MA: 15–44.

Axelrod R (1997) Advancing the art of simulation in the social sciences. In: Conte R, Hegselmann R, Terna P (eds.) *Simulating Social Phenomena.* Springer-Verlag, Berlin: 21–40.

Chen S-H (ed.) (2002) *Evolutionary Computation in Economics and Finance.* Physica-Verlag, Berlin.

Chen S-H (ed.) (2002) *Genetic Algorithms and Genetic Programming in Computational Finance.* Kluwer, Boston, MA.

Fogel L, Owens A, Walsh M (1966) *Artificial Intelligence through Simulated Evolution.* Wiley, New York, NY.

Koza J (1992) A Genetic approach to econometric modeling. In: Bourgine P, Walliser B (eds.) *Economics and Cognitive Science.* Pergamon Press, Oxford, UK: 57–75.

LeBaron B (1999) Building financial markets with artificial agents: Desired goals and present techniques. In: Karakoulas G (ed.) *Computational Markets.* MIT Press, Cambridge, MA.

Lucas R (1986) Adaptive behavior and economic theory. In: Hogarth R, Reder M (eds.) *Rational choice: The contrast between economics and psychology.* University of Chicago Press, IL: 217–242.

2 Key Survey/Review Articles

Arifovic J (1994) Genetic algorithms learning and the cobweb model. *J. Economic Dynamics and Control,* 18(1): 3–28.

Bullard J, Duffy J (1998) A model of learning and emulation with artificial adaptive agents. *J. Economic Dynamics and Control,* 22: 179–207.

Holland J, Miller J (1991) Artificial adaptive agents in economic theory. *American Economic Review,* 81(2): 365–370.

LeBaron B (2001) Evolution and time horizons in an agent based stock market. *Macroeconomic Dynamics,* 5: 225–254.

LeBaron B, Arthur W, Palmer R (1999) Time series properties of an artificial stock market. *J. Economic Dynamics and Control,* 23: 1487–1516.

3 Journals

Computational Economics

Intl. J. Intelligent Systems in Accounting, Finance and Management

J. Computational Intelligence in Finance

J. Economic Dynamics and Control

J. Evolutionary Economics

J. Financial Economics

J. Monetary Economics

4 Key International Conferences/Workshops

4.1 Economics

Computational Intelligence in Economics and Finance (CIEF)

Intl. Conf. Computing in Economics and Finance (CEF)

Intl. Conf. Economic Science with Heterogeneous Interacting Agents (ESHIA)

World Conference on Social Simulation (WCSS)

4.2 Agents

Annual Conference on Neuroeconomics

Intl. ESA Conf. Experimental Economics (Economic Science Association)

International Workshop on Agent-Based Approaches in Economic and Social Complex Systems (AESCS)

North American Association for Computational Social and Organizational Sciences (NAACSOS)

5 (Open Source) Software

`MASON`: Multi-Agent Simulator
http://cs.gmu.edu/ eclab/projects/mason/

`MATLAB`
http://www.mathworks.com/

`NetLogo`
http://ccl.northwestern.edu/netlogo/

`Repast`
http://repast.sourceforge.net/

`Sociodynamica`
http://atta.labb.usb.ve/Klaus/Programas.htm

`StarLogo`
http://education.mit.edu/starlogo/

`Swarm`
http://www.swarm.org/wiki/Main_Page

6 Data Bases

`COMPUSTAT`
https://www.compustatresources.com/support/index.html

`CRSP`
http://www.crsp.com/products/stocks.htm

`DatAnalysis`
http://www.deakin.edu.au/library/search/title/datanalysis.php

`Datastream`
http://www.datastream.com/

`Global Financial Data`
http://www.globalfinancialdata.com/

`Yahoo! Finance`
http://finance.yahoo.com/