Collaborative Computational Intelligence in Economics

Shu-Heng Chen

Abstract. In this chapter, we review the use of the idea of collaborative computational intelligence in economics. We examine two kinds of collaboration: first, the collaboration within the realm of computational intelligence, and, second, the collaboration beyond the realm of it. These two forms of collaboration have had a significant impact upon the current state of economics. First, they enhance and enrich the heterogeneous-agent research paradigm in economics, alternatively known as agent-based economics. Second, they help integrate the use of human agents and software agents in various forms, which in turn has tied together agent-based economics and experimental economics. The marriage of the two points out the future of economic research. Third, various hybridizations of the CI tools facilitate the development of more comprehensive treatments of the economic and financial uncertainties in terms of both their quantitative and qualitative aspects.

1 Introduction

Computational intelligence has been applied to economics for more than a decade. These applications can be roughly divided into two categories, namely, *agent-based computational economics* and *financial data mining*. Although in many such studies only one computational intelligence (CI) tool is involved, studies which apply more than one CI tool also exist and have become popular. In these studies, a few CI tools *work together* or *collaborate* with each other to perform a certain function. These studies are, therefore, examples of the use of *collaborative computational intelligence.* In this chapter, we shall provide a general review of collaborative computational intelligence in economics based on these studies.

There are three major sources that motivate the application of collaborative computational intelligence (CCI) to economics. The first source of stimulation

Shu-Heng Chen
Department of Economics, National Chengchi University
e-mail: chchen@nccu.edu.tw

C.L. Mumford and L.C. Jain (Eds.): Computational Intelligence, ISRL 1, pp. 233–273.
springerlink.com

comes from the new research paradigm in macroeconomics, which is known as the *heterogeneous-agent research paradigm*. This new research paradigm is an alternative to the conventional *representative-agent paradigm*, which has dominated the development of macroeconomics for about half a century [54]. In Section 2, we shall comprehensively review this development and point out its relevance to CCI.

The second source pertains to the recent attempt to integrate the *experimental economics* and *agent-based computational economics* [22, 27, 45, 85]. One main task associated with this integration is an inquiry into the relationship between human agents and software agents. Three relations have been mentioned in the literature, namely, *mirroring*, *competition*, and *collaboration*. In Section 3, we shall review this development and indicate how CCI can be applied along these lines.

The last source is the use of hybrid systems in *financial data mining* [16, 109]. Over the last two decades, we have evidenced the simultaneous use of multiple CI tools to build intelligent systems, including many financial ones. However, a general review of this development is beyond the scope of this chapter. Besides, related discussions may be available from other chapters. Therefore, to avoid redundancy and to use the limited size efficiently, in Section 4, we shall focus on only the few frequently used economic and financial hybrid systems in financial data mining. Section 5 will give the concluding remarks.

2 Heterogeneous Agents

Why is collaborative computational intelligence relevant to the study of economics? There is a straightforward answer: economic agents are *heterogeneous*, and their differences and interactions match the idea of CCI well. In this section, we shall see how computational intelligence has worked with the conventional approach in modeling a population of heterogeneous agents and their interactions. Basically, these types of collaboration can be differentiated into three levels, from the *macroscopic*, to the *microscopic*, to the *molecule* level. We shall first briefly state these three levels of collaboration (Section 2.1), and then elaborate on the significance of the collaboration at each level by highlighting existing research (Sections 2.2-2.4).

2.1 The Three Levels of Collaboration

From a *macroscopic* viewpoint, the population-based CI tools, such as *evolutionary computation* and *swarm intelligence*, have already encapsulated the idea of *population dynamics*. Therefore, they are readily applicable to model a population of heterogeneous economic agents. As we shall see in Section 2.2, there has already been a great deal of this kind of collaboration. On the other hand, from a *microscopic* viewpoint, we may also represent the heterogeneity of agents *individually* by different CI algorithms. In this case, these agents can be regarded as incarnations of different CI algorithms. These algorithms may be the same and may only differ in their control parameters, or else they may be fundamentally different. For example, in the former case, all agents are represented by genetic programming, while using different parameters of population size [36], or, for the latter case, some agents are represented by the *K-nearest-neighbors* (KNN) algorithm or general *instance-based learning* (IBL) algorithms, while some others are represented by Bayesian learning [21].

By moving further down to even more fine-detail level, referred to as the *molecule level*, we can regard each individual agent as being represented by more than one CI tool [67, 98]. In other words, the idea of *hybrid systems* is applied to model agents, and hence their heterogeneity can also be manifested in terms of different hybridization styles.

2.2 Macroscopic Level: Evolving Population

From the macroscopic viewpoint, the collaboration of computational intelligence with the conventional *cluster analysis* has developed the conventional economic agent engineering from the *N-type designs* into the *autonomous-agent designs*. This development is particularly evident in the *agent-based financial modeling*. To review this progress, let us first briefly review how agent-based economic and financial modeling have arisen (Section 2.2.1), and then consider the two different approaches in the designs of economic and financial agents (agent engineering) (Sections 2.2.2-2.2.3).

2.2.1 Agent-Based Economic Modeling

The rise of agent-based economics and finance can be considered to be a paradigm shift after long questioning, and even dissatisfaction with, the mainstream economics methodology built upon *representative agents*. [54] provides a lengthy discussion on this "troubling" concept. There are many reasons for going against the device of the representative agents, both from the empirical and theoretical aspects. The main empirical grounds are that there is ample empirical evidence to show that great heterogeneity and diversity exists at the micro level, from households, firms, traders, and other decision-makers. Nonetheless, a solid understanding of this diversity, such as the wealth distribution of households, the size distribution of firms, and the optimistic and pessimistic forecasting distributed among financial practitioners, is still lacking. Therefore, there is a need to search for a more suitable methodology in order to study the distributive behavior of an economy. Furthermore, given the great diversity at the micro level, the relationship between the macro (aggregates) and micro (individuals) becomes much more complex than that of which holds in the representative-agent economy. The exact relationship between the micro and macro has actually presented economic theorists with a new challenge. All these together motivated the formation of heterogeneous-agent approaches or the agent-based paradigm approach to economics in the 1990s.[1]

[1] There are a number of textbooks on macroeconomics written within this new background. See, for example, [5, 44].

Agent-based economics is an application of agent-based tools to economics. As with other agent-based models, the agent-based economics also begin with *agents*. This starting procedure is mainly composed of technical characterizations of economic agents, i.e., how to design economic agents. Hence, it is also called *economic agent engineering*. Economic agent engineering matters because, in general, it is expected that different designs of agents can result in different aggregate dynamics, even under the same institutional arrangements. Therefore, the agent-based economic model can serve as a tool to run a sensitivity analysis of a specific market or institution design to evaluate its possible performance. In this sense, it enables social scientists to have their own laboratory and to perform their own experiments as natural scientists do.

What follows are two very different designs of financial agents. The first one is called the *N-type design* (Section 2.2.2), whereas the second is called the *autonomous-agent design* (Section 2.2.3).

2.2.2 N-Type Designs

Let us now focus on the core of the agent-based financial markets, namely, financial agents and their design. In reality, financial agents can differ in many dimensions, ranging from expectations formation (beliefs), trading strategies, information exposure, risk attitudes,and wealth (investment scale), to the need for liquidity, etc. Given this high-dimensional heterogeneity, the essential question for financial agent engineering is to decide how much heterogeneity is to be reflected in the artificial markets. How much coarsely or finely do we want to differentiate these financial agents?

Before we examine the design of artificial financial agents, it is useful to recall what we have done for other artifacts. To name a few, the design of artificial ants (*ant algorithms*) was motivated by observing the behavior of real ants in a laboratory; the design of artificial bacteria (*bacterial algorithms*) was inspired by the microbial evolution phenomenon; the design of the artificial brain (*neural networks*, *self-organizing maps*) was motivated by the study of the real human brain; and the design of the evolutionary process (*evolutionary algorithms*) was inspired by real biological evolution. Generally speaking, the design of an artifact is, by and large, motivated and guided by the behavior of its counterpart in nature.

The design of artificial financial agents is no exception. It is highly motivated by observing how real financial agents behave. Empirical evidence accumulated since the late 1980s and early 1990s has shed new light on the forecasting behavior of financial agents. This empirical evidence was obtained through different kinds of surveys, such as questionnaires and telephone interviews, with financial specialists, bankers, currency traders, and dealers, etc. [50, 2]. The general findings from these abundantly established empirical data are two-fold. First, the data indicate that, by and large, there are two kinds of expectations existing in the market. The one which is characterized as a stabilizing force of the market is associated with a type of financial agent, called the *fundamentalist*. The one which is characterized as a destabilizing force is associated with another type of financial agent, called the *chartist*, *technical analyst* or *trend extrapolator*. Second, the proportion (micro-structure) of fundamentalists and chartists, also called the *market fraction*, is changing over time, which indicates that financial agents are adaptive. These empirical findings provide the initial direction for the early development of financial agent engineering. First, they suggest what rules to look at; second, they point out the significance of learning and adaptation.

Fundamentalists and chartists are concerned with two very different beliefs regarding the stock price dynamics. In a simple setting, they differ in terms of the mean-reverting speed of the stock price when it is mispriced (undervalued or overvalued). Fundamentalists tend to believe that the mispriced situation will soon be corrected, whereas chartists tend to believe that in the short run it will continue.

2-Type Design

To make what we say more precise, we generally denote the forecasting rule of a type-h agent as follows:

$$E_{h,t}[p_{t+1}] = f_{h,t}(p_t, p_{t-1}, \ldots), \tag{1}$$

where $E_{h,t}$ refers to the expectations of the type-h agent at time t. Equation (1) indicates the one-step ahead forecast. At the beginning, we start with a very general forecasting function $f_{h,t}$, which uses all the historical data on price up to the present. In addition, by considering that agents are adaptive, we allow the function to change over time and hence denote it by the subscript t.

For the fundamentalists ($h = f$) and chartists ($h = c$), their forecast rules, in a very simple setting, can be written as

$$E_{f,t}[p_{t+1}] = p_t + \alpha_f(p_t^f - p_t), \;\; 0 \leq \alpha_f \leq 1., \tag{2}$$

$$E_{c,t}(p_{t+1}) = p_t + \alpha_c(p_t - p_{t-1}), \;\; 0 \leq \alpha_c. \tag{3}$$

The idea of these two behavioral rules is that the fundamentalist has a *mean-reverting* belief, and his belief is characterized by a reverting coefficient (α_f), whereas the chartist has a trend-continuing belief, and his belief is characterized by an extrapolating coefficient (α_c). The magnitude of the reverting coefficient (α_f) measures the speed at which the fundamentalists expect the price to return to the fundamental one (p_t^f), whereas the magnitude of the extrapolating coefficient (α_c) expresses the degree to which chartists expect the past to change into the future.

3-Type Design

There is little doubt that the behavior of financial agents can be more complex than the two-type design. One obvious way to scale-up this design is to add more types of agents to the model so as to take into account a finer degree of heterogeneity of financial agents. This type of expansion is called the *N-type design*. For example, in a three-type design, one can further distinguish two kinds of chartists, namely, *momentum traders* and *contrarian traders*, or simply, *contrarians*. Like momentum traders, contrarians extrapolate past movements of the price into the future, but they follow the opposite of the trend. More precisely, their forecasting rule is as follows:

$$E_{co,t}(p_{t+1}) = p_t + \alpha_{co}(p_t - p_{t-1}), \quad \alpha_{co} \leq 0. \tag{4}$$

Contrarians consider that the price trend will finish soon, and will start to reverse. However, unlike fundamentalists, contrarians do not base their forecasts on the fundamental price, which they either do not know, or they do not care about.

The recent availability of more proprietary data has enhanced the transparency of the trading behavior of financial agents, including both individual and institutional investors. Empirical studies using such data have shown that individuals and institutions differ systematically in their reaction to past price performance and the degree to which they follow momentum and contrarian strategies. On average, individual investors are contrarian investors: they tend to buy stocks that have recently underperformed the market and sell stocks that have performed well in recent weeks [15]. With this empirical basis, financial agent engineering has already added the contrarians to the fundamentalist-chartist model, and popularized this three-type design.

Generalization of 2- and 3-Type Designs

Financial agent engineering can also be advanced by enriching the behavioral rules associated with each type of financial agent. This alteration may make financial agents more interdisciplinary. Considerations from different fields, including neural sciences, cognitive psychology, and statistics, can be incorporated into designs. For example, in behavioral finance, there is a psychological bias known as the "*law of small numbers*", which basically says that people underweight long-term averages, and tend to put too much weight on recent experiences (the recency effect). When equity returns have been high for many years, financial agents with this bias may believe that high equity returns are "normal". By design, we can take such bias into account. One way to do so is to add a *memory parameter* to the behavioral rules of our financial agents. This more general rule for contrarians is specified as follows:

$$E_{c,t}(p_{t+1}) = p_t + \alpha_c(1-\beta_c)\sum_{i=0}^{T}(\beta_c)^i(p_{t-i} - p_{t-i-1}), \quad 0 \leq \alpha_c, \quad 0 \leq \beta_c \leq 1. \tag{5}$$

$$E_{co,t}(p_{t+1}) = p_t + \alpha_{co}(1-\beta_{co})\sum_{i=0}^{T}(\beta_{co})^i(p_{t-i} - p_{t-i-1}), \quad 0 \geq \alpha_{co}, \quad 0 \leq \beta_{co} \leq 1. \tag{6}$$

The momentum traders and contrarians now compute a moving average of the past changes in the stock price and they extrapolate these changes into the future of the stock price. However, we assume that there is an exponential decay in the weights given to the past changes in the stock price. The parameters β_c and β_{co} can be interpreted as reflecting the memory of momentum traders and contrarians. If $\beta_c = \beta_{co} = 0$, momentum traders and contrarians remember only the last period's price change and they extrapolate this into the future. When β_c and β_{co} increase, the weight given to the price changes farther away in the past increases. In other words, the chartists' memory becomes longer.

The psychological bias mentioned earlier, therefore, corresponds to a small value of this memory parameter, and this "hypothesis" can actually be tested. In fact, by using the data for the S&P 500 index, one of the three major US stock market indices, from January 1980 to December 2000, [4] actually estimated a three-type agent-based financial market model, and found that contrarians have a longer memory than momentum traders when they form their forecast of the future price. Of course, this is just the beginning in terms of seeing how agent-based financial market models can be quantified so as to communicate with behavioral finance.

Adaptive Behavior

In the original fundamentalist-chartist model, learning does not exist. Agents who initially happen to be fundamentalists will continue to be fundamentalists and will never change this role, and likewise for chartists. As a result, the proportion (market fraction) of fundamentalists and chartists remains fixed. Nonetheless, this simplification underestimates the uncertainty faced by each trader. In general, traders, be they fundamentalists or chartists, can never be certain about the duration of the biased trend, since the trend can finish in weeks, months, or years. This uncertainty causes the alerted traders to review and revise their beliefs constantly. In other words, traders are *adaptive*.

Therefore, a further development of financial agent engineering is to consider an evolving micro-structure of market participants. In this extension, the idea of adaptive agents or learning agents is introduced into the model. Hence, an agent who was a fundamentalist (chartist) may now switch to being a chartist (fundamentalist) if he considers this switching to be more promising. Since, in the two-type model, agents can only choose to be either a fundamentalist or a chartist, modeling their learning behavior becomes quite simple, and is typically done using a *binary-choice model*, specifically, the *logit model* or the *Gibbs-Boltzmann distribution*.

The logit model, also known as the *Luce model*, is the main model used in the psychological theory of choice, and was proposed by Ducan Luce in 1959 in his seminal book, "*Individual Choice Behavior: A Theoretical Analysis*." Consider two alternatives f (fundamentalist) and c (chartist). Each will produce some gains to the agent. However, since the gain is random, the choice made by the agent is random as well. The logit model assumes that the probability of the agent choosing f is the probability that the profits or utilities gained from choosing f are greater than those gained from choosing c. Under a certain assumption for the random component of the utility, one can derive the following *binary logit model*:[2]

$$Prob(X = f, t) = \frac{\exp^{\lambda V_{f,t-1}}}{\exp^{\lambda V_{f,t-1}} + \exp^{\lambda V_{c,t-1}}}, \tag{7}$$

[2] The extension into the multinomial logit model is straightforward.

where $V_{f,t}$ and $V_{c,t}$ are the deterministic components of the gains from the alternatives f and c at time t. The parameter λ is a parameter carried over from the assumed random component. The logit model says that the probability of choosing the alternative f depends on its *absolute deterministic advantages*, as we can see from the following reformulation:

$$Prob(X = f,t) = \frac{1}{1 + \exp^{-(\lambda(V_{f,t-1} - V_{c,t-1})}}. \tag{8}$$

When applied to the agent-based financial models, these deterministic components are usually related to the temporal realized profits associated with different forecasting rules. So, in the two-type model, if V_f can be the temporal realized profits from being a fundamentalist, then V_c can be the temporal realized profits from being a chartist. In addition, there is a new interpretation for the parameter β, namely, the *intensity of choice*, because it basically measures the extent to which agents are sensitive to the additional profits gained from choosing f instead of c.

Market Maker Equation

The market fractions above then determine the market fraction of each type of agent in the market. For example, if $Prob(X = F) = 0.8$, it means that 80% of the market participants are fundamentalists and the remaining 20% are chartists. The asset price will be determined by this market fraction via the *market maker equation*.

$$p_t = p_{t-1} + \mu_0 + \mu_1 D_t \tag{9}$$

where

$$D_t = \sum_h w_{h,t} d_{h,t} = \sum_h Prob(X = h,t) d_{h,t}. \tag{10}$$

Equation (9) is the market maker equation, which assumes that the price is adjusted by the *market maker*, whose decision is in turn determined by the excess demand normalized by the number of market participants, D_t. D_t, in Equation (10), is a weighted average of the individual demand of each type of trader, weighted by the market fractions (7).

Risk Preference and Portfolio

The demand for assets of each type of trader is derived in a standard expected-utility maximization manner, which depends on the *risk preference* of the type-h agent. Risk preference is important because it is the main determinant of agents' portfolios, i.e., how agents' wealth is distributed among different assets. The classical *Markowitz mean-variance portfolio selection model* offered the first systematic treatment of asset allocation. Harry Markowitz, who later received the 1990 Nobel Prize in Economics for this contribution, assumes that investors are concerned only with the mean and variance of returns. This *mean-variance preference* has been extensively applied to modeling agents' risk preference since the variance of returns is normally accepted as a measure of risk.

In addition to the mean-variance preference, there are two other classes of risk preferences that are widely accepted in the standard theory of finance. These two correspond to two different attitudes toward risk aversion. One is called *constant absolute risk aversion* (CARA), and the other is called *constant relative risk aversion* (CRRA). When an agent's preference exhibits CARA, his demand for the risky asset (or stock) is independent of his changes in wealth. When an agent's preference exhibits CRRA, his demand for risky assets will increase with wealth in a linear way. Using a Taylor expansion, one can connect the mean-variance preference to CARA preferences and CRRA preferences. In fact, when the returns on the risky assets follow a normal distribution, the demand for risky assets under the mean-variance preference is the same of that under the CARA preference, and is determined by the *subjective-risk-adjusted expected return*.

$$d_{h,t} = \frac{E_{h,t}(\Delta p_{t+1})}{a_{h,t}V_{h,t}(\Delta p_{t+1})} = \frac{E_{h,t}(p_{t+1}) - p_t}{a_{h,t}V_{h,t}(\Delta p_{t+1})}, \tag{11}$$

where $\Delta p_{t+1} = p_{t+1} - p_t$, and $a_{h,t}$ is a risk aversion coefficient. The $E_{h,t}(p_{t+1})$ in the numerator of Equation (11) is given by Equations (2), (3) and (4), and $V_{h,t}$ in the denominator represents the perceived risk of the type-h agents. Further details of the formation of this subjective perceived risk can be found in the agent-based finance literature [42, 58].

Use of the N-Type Designs

While putting this N-type design into practical financial forecasting is still in its infancy stage, we have already seen some successful initial attempts in foreign exchange markets, which can be found in [43], a three-type design, and [75], a two-type design.

2.2.3 Autonomous-Agent Designs

So far, all the types and rules of financial agents are given at the beginning of the design, and what financial agents can do is to choose among these different types and rules based on their past experiences. The N-type design has characterized a major class of agent-based financial markets. However, this way of doing things also severely restricts the degree of autonomy available for financial agents. First, they can only choose how to behave based on what has been offered; secondly, as a consequence, there will be no new rules available unless they are added outside by the designers. If we want our artificial financial agents to behave more like real financial agents, then we will certainly expect that they learn and discover *on their own*. Therefore, as time goes by, new rules which have never been used before and have not been supplied by the designer may be discovered by these artificial agents inside the artificial world.

Genetic Algorithms

Designing artificial agents who are able to design on their own is an idea similar to John von Neumann's *self reproducing automata*, i.e., a machine which can reproduce itself. This theory had a deep impact on John Holland, the father of the *genetic algorithm*indexgenetic algorithms. Under von Neumann's influence, Holland had devoted himself to the study of a general-purpose computational device that could serve as the basis for a general theory of automata. In the 1970s, he introduced the genetic algorithm, which was intended to replace those ad hoc learning modules in contemporary mainstream AI. Using genetic algorithms, Holland could make an adaptive agent that not only learned from experience but could also be spontaneous and creative. The latter property is crucial for the design of artificial financial agents. In 1991, Holland and John Miller, an economist, published a sketch of the artificial adaptive agent in the highly influential American Economic Review. This blueprint was actually carried out in an artificial stock project in 1988 in the Santa Fe Institute [82, 10].

Sante Fe Institute Artificial Stock Market

Armed with GAs, the *Santa Fe Artificial Stock Market* (SFI-ASM) considers a novel design for financial agents. First, like many N-type designs, it mainly focuses on the forecasting behavior of financial agents. Their trading behavior, as depicted in Equation (11), will depend on their forecasts of the price in the next period. Second, however, unlike the N-type designs, these agents are not divided into a fixed number of different types. Instead, the forecasting behavior of each agent is "customized" via a GA. We shall be more specific regarding its design because it provides us with a good opportunity to see how economists take advantage of the increasing computational power to endow artificial decision makers with a larger and larger degree of autonomy.

In the SFI-ASM, each financial agent h uses a linear forecasting rule as follows:

$$E_{h,t}(p_{t+1}) = \alpha_{h,t} + \beta_{h,t} p_t. \tag{12}$$

However, the coefficients $\alpha_{h,t}$ and $\beta_{h,t}$ not only change over time (time-dependent), but also are state-dependent. That is, the value of these two coefficients at time t will depend on the state of the economy (market) at time t. For example, the recent price dynamics can be an indicator, so, say, if the price has risen in the last 3 periods, the financial agent may consider lower values of both α and β than otherwise. The price dividend ratio can be another indicator. If the price dividend ratio is lower than 50%, then the financial agent may want to take a higher value of β than if it is not. This state-dependent idea is very similar to what is known as *classification and regression trees* (CART) or *decision trees*, a very dominant approach in machine learning.

Therefore, one simple way to think of the artificial agents in the SFI-ASM is that they each behave as machine-learning people who use *regression trees* to forecast the stock price. At each point in time, the agent has a set of indicators which help him to decompose the state of the economy into m distinct classes, $(A^1_{h,t}, A^2_{h,t}, ..., A^m_{h,t})$, and corresponding to each of the classes there is an associated linear forecasting model. Which model will be activated depends on the state of the market at time t, denoted by S_t. Altogether, the behavior of the financial agent can be summarized as follows:

$$E_{h,t}(p_{t+1}) = \begin{cases} \alpha^1_{h,t} + \beta^1_{h,t} p_t, & if \;\; S_t \in A^1_{h,t}, \\ \alpha^2_{h,t} + \beta^2_{h,t} p_t, & if \;\; S_t \in A^2_{h,t}, \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \alpha^m_{h,t} + \beta^m_{h,t} p_t, & if \;\; S_t \in A^m_{h,t}. \end{cases} \tag{13}$$

A few remarks are added here. First, the forecasting rule summarized above is updated as time goes by, as we keep the subscript t there. So, agents, in this system, are learning over time with a regression tree, or they are using a time-variant regression tree, in which all the regression coefficients and classes may change accordingly with the agents' learning. Second, agents are completely heterogeneous as we also keep the subscript h above. Therefore, if there are N financial agents in the markets at each point in time, we may observe N regression trees, each of which is owned and maintained by one individual agent. Third, however, the forecasting rules introduced in the SFI-ASM are not exactly regression trees. They are, in fact, *classifier systems*.

Classifier System

A classifier system is another of John Holland's inventions in the late 1970s. This system is similar to the Newell-Simon type of expert system, which is a population of if-then or condition-action rules. The conventional expert systems are not able to learn by themselves. To introduce adaptation into the system, Holland applied the idea of market competition to a society of if-then rules. A formal algorithm, known as the *bucket-brigade algorithm*, credits rules generating good outcomes and debits 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. Therefore, a GA is applied to evolve them and to discover new rules.

This autonomous-agent design has been further adopted in many later studies. While most studies continuously carried out this task using genetic algorithms[3], a few studies also used other population-based learning models, such as evolutionary programming and genetic programming.

[3] A lengthy review of this literature can be found in [23].

Genetic Programming and Autonomous Agents

The development from the few-type designs to the many-type designs and further to the autonomous-agent designs can be considered to be part of a continuous effort to increase the collective search space of the forecasting function $E_{h,t}$, from finite to infinite space, and from parametric to semi-parametric functions. The contribution of genetic programming (GP) to this development is to further extend the search space to a infinite space of non-parametric functions, whose *size* (e.g., the dimensionality, the cardinality or the number of variables used) and *shapes* (for example, linearity or non-linearity, continuity or discontinuity) have to be determined, via search, simultaneously. This way of increasing the degree of autonomy may not contribute much to the price dynamics, but can enrich other aggregate dynamics as well as the behavior at the individual level. As we shall see below, the endogenous determination of the size and shape of $E_{h,t}$ provides us with great opportunities to see some aspects of market dynamics which are not easily available in the N-type designs or other autonomous-agent design.

The first example concerns *the sophistication of agents* in market dynamics. The definition and operation of GP rely on a specific language environment, known as LIST Programming (LISP). For each LISP program, there is a tree representation. The number of nodes (leaves) or the number of depths in the LISP trees provides one measure of complexity in the vein of the *program length*. This additional observation enables us to study not just the heterogeneity in $E_{h,t}$, but also the associated complexity of $E_{h,t}$. In other words, genetic programming can not only distinguish agents by their forecasts, as the N-type designs did, but further delineate the differentiation according to the agents' sophistication (complexity). Must the survival agents be sophisticated or can the simple agents prosper as well?

One interesting hypothesis related to the above inquiry is the *monotone hypothesis*: the degree of traders' sophistication is an increasing function of time. In other words, traders will evolve to be more and more sophisticated as time goes on. However, this hypothesis is rejected in [33]. They found that, based on the statistics on the node complexity or the depth complexity, traders can evolve toward a higher degree of sophistication, and at some point in time, they can be simple as well.

The second example concerns the capability to distinguish the information from noise. As we mentioned earlier, the variables recruited in the agents' forecasting function are also endogenously determined. This variable-selection function allows us to examine whether the smart picking of these variables is crucial for survival. In particular, the hypothesis of the extinction of noisy traders says that traders who are unable to distinguish information from noise will become extinct. [34] test this hypothesis. In an agent-based artificial market, they supplied traders with both informative and noisy variables. The former include prices, dividends and trading volumes, whereas the latter are just series of pseudo random numbers. Their simulation shows, as time goes on, that traders who are unable to distinguish information from noise do have a tendency to decline and even become extinct.

2.3 *Microscopic Level: Heterogeneity in Intelligence*

2.3.1 Bounded Rationality and Intelligence Quotient

At the microscopic level, collaborative computational intelligence has shown its relevance to modeling *bounded rationality*. Computational intelligence can be collaborated to address bounded rationality because different CI tools themselves may already demonstrate different degrees of rationality or intelligence. Having said that, we are aware of the measurement problems pertaining to rationality or intelligence. Certainly, so far, there is no formal measure of rationality, and whether it can be positively related to the *intelligence quotient* (IQ) is also unclear[4], even though the latter is frequently used as a proxy for the former.[5] In addition, our experience that smart people are not immune from doing dumb things further casts doubt on the connection between the two.[6] Needless to say, the study of human intelligence is still an open-ended on-going body of research. The current research trend in empirical economics, however, has attempted to make the behavior of bounded-rational agents transparent or observable using real-world data. In addition, bounded rationality is frequently used as an input in models since it may generate different predictions or outcomes.

Therefore, despite the lack of an acceptable measure of rationality, the current trend in economic research forces us to ask how computational intelligence can help us building economic models of bounded rationality or building bounded-rational agents. The simplest answer is that CI tools can help us to model the *learning* or *adaptive behavior* of bounded-rational agents. A huge economic literature has already documented this development.[7] Almost all major CI tools have been applied to model the learning and adaptive behavior of economic agents, that includes reinforcement learning, instance-based learning, regression trees, Bayesian learning, artificial neural nets, fuzzy logic, and evolutionary computation [23].

However, these studies have been frequently criticized as *ad hocry* in terms of the choice of a specific CI tool. Hence, to move forward, the research question to address is: can we have a theory or an acceptable practice to guide us in the choice of CI tools when modeling the adaptive economic agents? This question has motivated

[4] Despite their incurring criticisms, some empirical studies support a positive correlation between IQ and income. While the correlation coefficient is often found to be less than 0.5, it may increase with age to some extent [57, 61].

[5] The most famous example is the device of the *zero-intelligence agents* introduced in [52]. To motivate this design, [52] raised the issue: *how much intelligence is required of an agent to achieve human-level trading performance?* The zero-intelligence agent, based on the design of [52], is a *randomly behaved agent*, who needs no memory, no learning, and no strategic playing. It is a kind of naive agent, who just randomly bids or randomly asks. It was found that these randomly behaved agents are sufficiently able to replicate the market efficiency achieved by human agents.

[6] That is why we frequently see books like [48].

[7] Among many available textbooks, [89] is the first one which introduces the materials of computational intelligence to economists. [47] also has a section introducing the use of computational intelligence to build learning models.

a number of research directions. The one which is related to *experimental economics* will be addressed in Section 3. In this subsection, we address the one directly related to *controlling the degree of intelligence or smartness*. However, before that, let us make a distinction between the two.

Certainly, models of learning and adaptation should be a part of bounded rationality, but not the whole. What has been generally neglected in the past applications of CI tools to learning is that *economic agents are not equally smart*, as they have different IQ, EQ, or whatever Q.[8] Even though they are all learning, it does not mean that they are learning under the same cognitive constraints or under the same mental capacity. In fact, current *behavioral genetics* enhance our understanding of the heterogeneity in human cognitive ability, that includes the ability to memorize, to search, to learn, to perceive, and to socialize [76, 84]. A large proportion of the variance in cognitive abilities can be attributed to genetics. Therefore, the difference in *genome* may need to be incorporated into our applications of CI tools, and agents with different innate cognitive abilities are expected to be equipped with different CI tools.

2.3.2 Heterogeneity Characterized by Different CI Tools

Fortunately, taxonomies of CI tools based on the degree of cognitive constraints are possible, while not perfect.[9] For example, reinforcement learning models tend to be regarded as less sophisticated than evolutionary computational models. Therefore, a market composed of agents with heterogeneity in intelligence can be considered to be a market composed of some "less smart" agents, whose adaptive behavior is driven by reinforcement learning, and some "smart" agents, whose adaptive behavior is driven by evolutionary computation. With this mixture, a few CI tools can *interact* with each other in the same economic environment. This defines the first kind of application of CCI at the micro level.[10]

[21] is probably the first study of this kind. In a context of agent-based artificial stock markets, [21] considers three different types of agents. The first is the momentum trader (chartist), whose forecasting behavior is a special case of Equation (3) when α_c is set to 1 and becomes [11]

$$E_{c,t}(p_{t+1}) = p_t + (p_t - p_{t-1}). \tag{14}$$

[8] Of course, IQ as an important causal determinant of decision making is not just neglected in economics, but is neglected in all social sciences [69].

[9] Both [17] and [45] give a comprehensive treatment of various learning algorithms.

[10] While various computational intelligence tools are often compared or compete in the financial engineering domains, such as financial forecasting, trading, etc.. There has been little work on comparing their behavior in agent-based economic modeling. The difference between the two study styles is that in the former case the competition or tournament is conducted without interaction or feedback, whereas in the latter case this mechanism is presented.

[11] See Section 2.2.2 for a detailed description of momentum traders.

This type of momentum trader is naive in the sense that they continuously believe that what they experience today regarding the price change will remain unchanged tomorrow. Given this naive momentum trader, they also introduced two sophisticated type of agents, namely, *empirical Bayesian traders* and *K-nearest-neighbor traders*. Both *empirical Bayesian traders* and *K nearest neighbors* (KNN) are active members of the CI toolkit.

Bayesian Learning

Before the advent of computational intelligence in the early 1990s, Bayesian learning was the dominant learning model used by economists. Economists have a strong preference for Bayesian learning partially because in spirit it is consistent with optimization. The optimality of Bayesian learning has been well established in *statistical decision theory*. It has a lot of variants and applications in regard to the economic modeling of learning. The two most popular ones are *Kalman filtering* and *recursive least squares*.[12]

As a Bayesian, the trader forecasts p_{t+1} using his posterior distribution (belief) of p_{t+1}, denoted by $f_{t+1}(p \mid x)$. $f_{t+1}(p \mid x)$ is the trader's updated subjective belief of the distribution of the price p_{t+1} after receiving the state information x at time t. The updating formula is the famous *Bayes rule*:

$$f_{t+1}(p \mid x) = \frac{f_t(p)h_t(x \mid p)}{\int f_t(\bar{p})h_t(x \mid \bar{p})d\bar{p}} \tag{15}$$

The Bayesian trader will then forecast using the posterior mean:

$$E_{b,t}(p_{t+1}) = \int p f_{t+1}(p \mid x), \tag{16}$$

where $E_{b,t}$ refers to the prediction made by the Bayesian trader at time t. Intuitively speaking, the Bayesian trader has a set of possible predictions (hypotheses) $S_t = \{p^e_{t+1}\}$, and not just a single degenerated prediction (hypothesis) p^e_{t+1}. The possibility of each of the possible predictions in the set S_t is governed by the posterior distribution (15). It is now clear that Bayesian traders need to have a greater mental capacity to first keep a set of hypotheses and then to deal with possibly very demanding computations involved in (15) and (16).[13]

There is, however, a way to reduce this very demanding work. With the assumption of the multivariate normal distribution, the entire updating of the posterior distribution can be reduced to the updating of two parameters only, namely, the mean and the variance. In this case, we have the familiar Kalman filtering. By denoting these expectations by $E_{k,t}$, then

[12] See [89] and [47] for details.

[13] It has been argued that the inability of humans to produce consistent and reliable probability and preference judgments may explain why Bayesian decision theory fails in view of this lack of necessary inputs.

$$E_{k,t}(p_{t+1}) = E_{k,t-1}(p_t) + k_t(p_t - E_{k,t-1}(p_t)), \tag{17}$$

where k_t is the *Kalman gain* at time t.

Alternatively, one can also simplify the possible messy computation by using the so-called empirical Bayes.[14] The empirical Bayesian trader basically behaves like a Bayesian, except that the posterior distribution is built upon an empirical rather than a subjective distribution. This simplification requires traders to "memorize" all of the association between the state information x and the price so that they can replace the posterior distribution $f_{t+1}(p \mid x)$ simply by using the most recent histogram. As we shall see below, this simplification connects the empirical Bayes to the K nearest neighbors, to which we now turn.

K Nearest Neighbors

KNN differs from the conventional time-series modeling techniques. The 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:

$$p_{t+1} = f(p_t, p_{t-1}, ..., p_{t-m}) + \varepsilon_t = f(\mathbf{P^m_t}) + \varepsilon_t \tag{18}$$

by using all of the information up to t, i.e., $\mathbf{P^m_s}, \forall s \leq t$, where the estimated function $\hat{f}$ is assumed to hold for every single point in time. As a result, what will affect p_{t+1} most is its immediate past $p_t, p_{t-1}, ...$ under the law of motion estimated by all available samples.

For KNN, while what affects p_{t+1} most is also its immediate past, the law of motion is estimated *only* with *similar* samples, and *not all* samples. The estimated function $\hat{f}_t$ is hence assumed to only hold for that specific point in time. To facilitate the discussion, we introduce the following notations.

$$\mathbf{P}^m_1, \mathbf{P}^m_2, ..., \mathbf{P}^m_T, \;\; \mathbf{P}^m_t \in \mathbf{R}^m, \;\; \forall\, t = 1, 2, ..., T \tag{19}$$

$$\mathbf{P}^m_t \equiv \{p_t, p_{t-1}, ..., p_{t-m}\}, \;\; p_{t-l} \in \mathbf{R}, \forall\, l = 0, 1, ..., m-1. \tag{20}$$

$\mathbf{P}^m_t$ is a windowed series with an immediate past of m observations, also called the m-history. Equation (19), therefore, represents a sequence of T m-histories which are derived from the original time series, $\{p_t\}^T_{t=-m+1}$, by moving the m-long window consecutively, each with one step.

KNN forms a cluster based on each $\mathbf{P}^m_t$, $\mathcal{N}(\mathbf{P}^m_t)$, as follows.

$$\mathcal{N}(\mathbf{P}^m_t) = \{s \mid Rank(d(\mathbf{P}^m_t, \mathbf{P}^m_s)) \leq k, \forall s < t\}, \tag{21}$$

[14] See [20] for a fine overview of empirical Bayes, and also [19] for an in-depth treatment. The BUGS software provides an implementation of empirical Bayes methods using *Markov Chain Monte Carlo* [51]. The software is available from http://www.mrc-bsu.cam.ac.uk/bugs.

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

$$\{\mathbf{P}_s^m \mid s < t\} \tag{22}$$

from the closest to the farthest.

Then, by assuming a functional relation, f, between p_{s+1} and $\mathbf{P}_s^m$ and using only the observations associated with $\mathscr{N}(\mathbf{P}_t^m)$ to estimate this function f_t [15], one can construct the tailor-made forecast for each p_t,

$$E_{knn}(p_{t+1}) = \hat{f}_t(\mathbf{P}_t^m). \tag{23}$$

In practice, the function f used in (23) 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,

$$E_{knn}(p_{t+1}) = \frac{\sum_{s \in \mathscr{N}(\mathbf{P}_t^m)} p_{s+1}}{k}, \tag{24}$$

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. KNN can also be viewed as another kind of empirical Bayes since Equation (24) can be related to the posterior mean (16).

Embodiment: Game-Theoretic CCI

The efficient market hypothesis implies that there are no profitable strategies, and hence learning, regardless of its formalism, does not matter. As a result, the three types of traders, momentum traders, empirical Bayesian and k-nearest-neighbor traders should behave equally well, at least in the long run. However, when the market is not efficient, and learning may matter, it is expectedthat smarter agents can take advantage of dumber agents. In their experiments, [21] found that momentum traders, who never learn, performed worst during the transition period when market is not efficient. Furthermore, the empirical Bayesian traders was also outperformed by the KNN traders. While the two types of traders started learning at the same time and competed with each other to discover the true price, evidently the KNN traders were able to exploit predictability more quickly than the empirical Bayesian traders.

[21] points to a new style of application of CCI to economics, namely, using an agent-based environment to allow for a more vivid competition of different CI tools, each of which is to represent an opportunity-seeking trader with different degrees

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

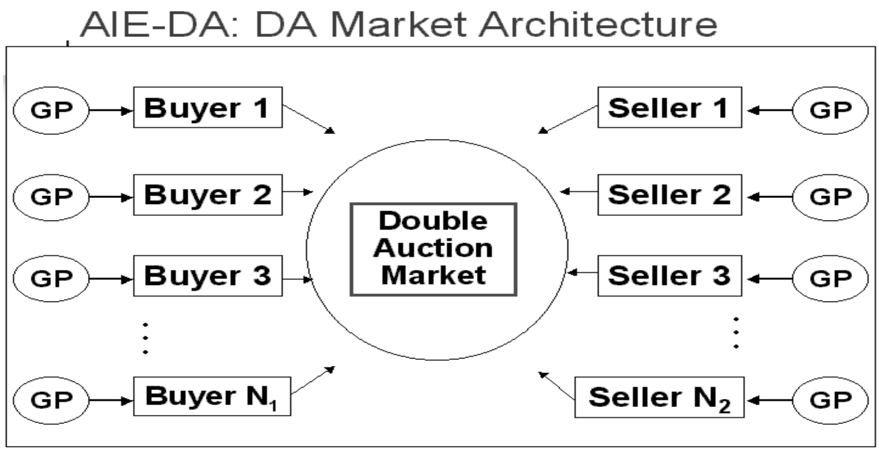

Fig. 1 Double Auction Market

of smartness. This style of application is not the same as a general forecasting tournament, in which a number of CI tools also compete with each other in forecasting a given time series. The key difference between the two styles of application lies in the *complex interacting effects* among these competing CI tools, which obviously exist in the former style, but not the latter. Put alternatively, [21] demonstrates an embodied game-theoretic environment for a set of CI tools, which may be coined as game-theoretic CCI.

2.3.3 Heterogeneity Characterized by Different Parameters

In addition to using different CI tools to characterize heterogeneity in intelligence, it is also possible to use different parameters of *the same CI tool* to distinguish different degrees of smartness. In this case, instead of the interaction of several different CI tools, what we observe is the interaction of the same CI tools, but which differ owing to different parameters.

In a study of the agent-based double auction market, [36] examine how the co-evolution of agents' strategies will change with the agents' level of smartness, and the associated micro-and-macro correspondence. Their agent-based double auction market is simulated with genetic programming (GP) (Figure 1). While GP enables all agents to learn, different values of control parameters of GP may imply different levels of smartness of agents. In this study, the parameter *population size* is used to characterize different degrees of smartness.

Using population size to characterize or approximate the degree of smartness can be justified as follows. Population size is positively associated with the search intensity. A larger search intensity can imply a higher performance.[16] In this vein, they vary their GP traders so as to have different population sizes, ranging from

[16] It is not always so. See [24].

10, 20,..., to 50. A smaller population size assumes a lower degree of smartness, whereas a larger population size implies a higher degree of smartness.

It is found that, other things being equal, increasing the intelligence of *individual traders* can contribute positively to the realized social welfare, a measure of market efficiency. Nevertheless, it is also found that, other things being equal, increasing the *number of intelligent traders* can exert an negative influence on the realized social welfare. These findings, therefore, suggest an interesting implication: if the increase in the number of intelligent traders is inevitable, then the increase in social welfare can be made possible only if all intelligent traders become smarter.

The significance of the degree of smartness is pursued in [25]. In the context of an agent-based artificial stock market, [25] address whether agents with different degrees of smartness may result in different wealth. This brings us closer to the original concerns of psychometricians mentioned in Section 2.3.1. In this study, artificial traders are all modeled by genetic algorithms (GA). They use a GA to do the forecasting, and then use it again to engage in portfolio optimization. By varying the control parameter of the GA, [25] are able to design traders with different levels of intelligence. In this case, the chosen control parameter is the *size of the validation window*, and this choice can be justified as follows.

In machine learning literature, it is very common to divide our data into three parts, namely, the training set, validation set, and testing set. The purpose of the validation sample is to prevent the trained model from being subjected to over-learning or over-fitting. In the environment of [25], it can be shown that the size of the validation window can affect the forecasting accuracy of the model constructed, which in turn will affect the quality of the portfolio decision. Through agent-based simulation, they, therefore, show that the agents' degree of smartness can positively affect their wealth share. Not surprisingly, the smarter they are, the wealthier they become.

2.3.4 Heterogeneity across Societies

Sections 2.3.2 and 2.3.3 are both concerned with an economy composed of agents with different degrees of smartness. These kinds of application can then examine how these different degrees of smartness can contribute to the resultant heterogeneity in economic performance. Therefore, they provide us with replications or predictions of the correlation between IQ and performance in a *social* context. One can further ask, to what extent, the *institutional design* can eliminate or minimize the impact of the heterogeneity in intelligence on the heterogeneity in economic performance, for example, income inequality [79].

However, one may also be concerned with the comparisons between different economies or different groups of agents. For example, [72, 71] provide rich resources on the comparative studies of IQ among different countries and races. On the other hand, differences in individuals' behavior among different societies can also be attributed to the *culture* factor. The recent path-breaking studies in this area can be found in [55, 56]. Using experimental results from the *ultimatum bargaining games*, [55] is able to show that "economic decisions and economic reasoning may be heavily influenced by cultural differences–that is, by socially transmitted rules about how to behave in certain circumstances (economic or otherwise) that may vary from group to group as a consequence of different cultural evolutionary trajectories." (Ibid, p. 973)

With these backgrounds, it is useful to distinguish different groups of agents by using different CI tools. For example, one can use reinforcement learning to model a group of agents, and use genetic algorithms to model another group of agents. One may also try the same CI tool but with different control parameters to characterize two different societies, as what we do in Section 2.3.3. Then by putting these two groups of agents separately in the same environment, such as the ultimatum bargaining game or stock market, one can compare the performance of the two different groups. In this way, the possible impacts of genetics and cultures upon the collective performance can be tackled. This research direction can be exemplified by the following few examples.

Minority Games

[90] addressed the traffic-flow problem in the context of games. This issue concerns the most efficient distribution of the road space among drivers, characterized by the travel time among different paths among drivers connecting the same origin with the destination being identical. The intriguing part of this issue is: can we achieve this goal in a bottom-up manner without the top-down supervision? [90] explored the possibility by assuming that each driver learns how to choose the paths by means of reinforcement learning. Several different versions of reinforcement learning have been attempted. They differ in one key parameter, *learning speed* or *the degree of forgetting*. It has been found that the allocative efficiency of roads is not independent of this parameter. In other words, unless the learning speed is tuned correctly, there is no guarantee that drivers will necessarily coordinate their use of roads in the most efficient way, and congestion can happen all the time.

The congestion problem, also known as *minority games*, originates from the famous *El Farol problem*, which was first studied by [9]. The problem concerns the attendance at the bar, El Farol, in Sante Fe. Agents' willingness to visit the bar on a specific night depends on their expectations of the attendance at that night. An agent will go to the bar if her expected attendance is less than the capacity of the bar; otherwise, she will not go. [9] showed that the time series of attendance levels seems to always fluctuate around an average level of the capacity of the bar. However, agents in [9] reason with *a fixed set of models*, deterministically iterated over time. Discovering new models is out of the question in this set-up.

[49] replace this fixed set of rules with a class of *auto-regressive* (AR) models. Furthermore, the number of lag terms and the respective coefficients are revised and renewed via *evolutionary programming* (EP). The introduction of EP to the system of AR agents has a marked impact on the observed behavior: the overall result is one of large oscillations rather than mild fluctuations around the capacity.

[90] and [49] together show that there is no guarantee that agents with *arbitrary learning algorithms*, characterizing different cultures, habits, routines, or IQ, can

coordinate well to avoid congestion and maximize social efficiency, and the *coordination limit* is affected by the IQ or cultures of society, which are characterized by various computational intelligence tools.

2.4 Molecule Level: Hybridization

The idea of hybrid systems is also employed to build individual agents. In this case, each individual is represented by more than one CI tool, and is an incarnation of a specific style of CCI.

Evolutionary Artificial Neural Nets

[67] provides the first application of this kind, and, in this case, the specific style is the *evolutionary artificial neural net*. In the context of the SFI artificial stock market, the financial agents are required to solve the portfolio optimization problem, which involves the distribution of the savings into risky and riskless assets, something which is similar to Equation (11). Equation (11) is a typical two-stage decision, i.e., the forecast decision is made before the investment decision, but [67] considered a reduced one-stage decision. The mapping is, therefore, directly constructed from the information available at time $t-1$ to the optimal portfolio at time t, $y_{h,t}$. More precisely, the financial agents are first represented by an artificial neural network, or a feedforward neural network with one hidden layer, to be exact.

$$y_{h,t} = h_2(w_0 + \sum_{j=1}^{l} w_j h_1(w_{0j} + \sum_{i=1}^{p} w_{ij} x_{i,t-1})) \tag{25}$$

The information set, $\{x_i\}_{i=1}^{p}$ includes past dividends, returns, the price/dividend ratio, and trend-following technical trading indicators. This population of investment decision rules (over all agents) is then evolved with *genetic algorithms* to symbolize the evolutionary learning of financial agents.

Genetic Fuzzy Classifier System

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 perceptions, 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 processes. The 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 agents' adaptive behavior is *crisp*. [98] made progress in this direction by using the *genetic-fuzzy classifier system* (GFCS) to model traders' adaptive behavior in the SFI-like artificial stock market.

[98] considers a fuzzy extension of the forecasting function (13). In Equation (13), each forecasting rule has two coefficients, the constant term (α) and the

slope (β). Without any augmentation, these forecasting rules are simply linear, and cannot be expected to work well. The original SFI-ASM made them non-linear by making these two coefficients *state dependent* via the classifier system. However, the two coefficients are crisp. [98] applies the Mamdani style of fuzzy rules to make them fuzzy. As an illustration, the Mamdani style of a fuzzy if-then rule is:

> If x is "A", then y is "B",

whereas the input set "A" and the output set "B" are both fuzzy. In [98], this application becomes something like:

> If $\frac{p_t}{MA(5)}$ is " low", then α is "moderately high", and β is "moderately high".

Obviously, the terms "low", "high", "moderately low", and "moderately high" are all linguistic variables, and they are represented by the respective membership functions. The state variable is $p_t/MA(5)$, where $MA(5)$ is the moving average of the price over the last five periods. So, this rule compares the current price with the 5-day moving average, and if the ratio is low enough, then both α and β will be moderately high. Of course, the above fuzzy forecasting rule can easily be extended to include more variables. For example,

> "If $\frac{p_t \times r_t}{d_t}$ is high and $p_t/MA(5)$ is moderately low and $p_t/MA(10)$ is moderately high and $p_t/MA(100)$ is low and $p_t/MA(500)$ is high, then α is "moderately low", and β is "high".

$p_t r_t/d_t$ reflects the current price in relation to the current dividend and it indicates whether the stock is above or below the fundamental value at the current price. The inclusion of this information makes agents behave like fundamentalists. The remaining four state variables indicate whether the price history exhibits a trend or similar characteristic. The inclusion of this information makes agents behave more like chartists. Therefore, by combining these state variables, the financial agents may choose to behave more like fundamentalists or more like chartists.[17]

3 Human and Software Agents

We have now reviewed how the idea of CCI enhances the heterogeneous-agent research paradigm at the macro, micro and molecule levels. In addition to that, the idea of CCI also plays an important role in the recent efforts made by economists to overarch agent-based computational economics (ACE) and experimental economics. It has been argued in many instances that agent-based simulation should be integrated with experiments using human subjects, for example, [45], [60] and [74]. The relationship between agent-based computational economics and experimental

[17] This design is not the 2-type design as we see in Section 2.2.2.

economics is, in essence, a relationship between *human agents* and *software agents*. The literature has already demonstrated three possible ways of closely relating ACE to experimental economics, namely, *mirroring*, *competition* and *collaboration*. They appear in the literature in chronological order.

3.1 Mirroring

The early ACE studies are clearly motivated by using software agents to mimic the behavior of human agents observed in the laboratory. The famous *Turing test* serves as the best illustration. [8] point out that the development of social science theories can be likened to the task of building a computer to mimic human behavior, or equivalently, to building a computer that will pass the Turing test in the range of behavior covered by the theory. Thus, a social science theory can be deemed to be successful when it is no longer possible for a computer judge to tell the difference between behavior generated by humans and that generated by the theory (i.e., by a machine).

3.1.1 Mirroring with Genetic Algorithms

In this regard, the two CI tools, namely, genetic algorithms and genetic programming are frequently used to build software agents such that their collective behavior can mirror the laboratories with human subjects. [6] pioneered this research direction. [6] applied two versions of GAs to study market dynamics in a *cobweb 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 involving 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, [70] 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. Other similar studies to justify the use of genetic algorithms to mirror human experiments include [7].

3.1.2 Mirroring with Genetic Programming

The application of genetic programming to the cobweb model started from [30]. [30] 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 can also 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 [6], does show up quite often in 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 [6]. Nevertheless, GP-based learning can quickly coordinate agents' beliefs so that the emergence of price euphoria is only temporary. Furthermore, unlike [6], [30] did not use the election operator. Without the election operator, the rational expectations equilibrium is exposed to potentially persistent perturbations due to the agents' adoption of the new, but untested, rules. However, what shows up in [30] 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 [30] was further tested with two complications. In the first case, [31] 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, and [32] then tested whether the market could restore the rational expectations equilibrium. The answer to the first experiment is generally negative, i.e., speculators do not enhance the stability of the market. On the contrary, they do 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. [32] showed that GP-based adaptive agents could detect the shift in the demand curve and adapt to it. 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.

In addition to genetic algorithms, genetic programming is also extensively applied to build systems of software agents which are able to replicate the laboratory results with human subjects. [26] studied bargaining behavior observed in the double-auction laboratory markets with human subjects. All buyers and sellers in [26] are *artificial adaptive agents*. Each artificial adaptive agent is built upon *genetic programming*. The architecture of genetic programming used is what is known as *multi-population genetic programming* (**MGP**). Briefly, they viewed or modeled

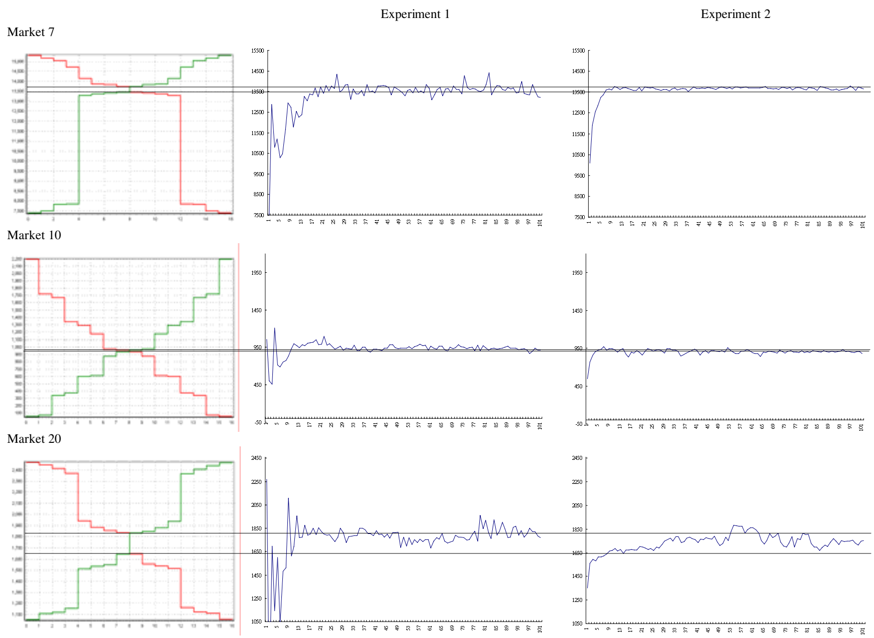

Fig. 2 Agent-Based Double Auction Market Simulation with GP Agents. The three markets presented here are selected and adapted from [26], Fig. 4

an agent as *a population of bargaining strategies*.[18] Genetic programming is then applied to *evolving* each population of bargaining strategies. In this case, a society of bargaining agents consists of many populations of programs. This architecture is shown in Figure 1.

In Figure 2, there are three plots. The leftmost plot is the market with its equilibrium price or equilibrium price interval. The middle plot and the rightmost plot are the time series of the GP simulations with different parameters. As we can see from Figure 2, the three markets presented here either have a unique equilibrium price (Market 10) or a tight equilibrium interval (Markets 7 and 20). Market prices in this case quickly move toward the equilibrium price (or price interval), and then slightly fluctuate around there. This result is basically consistent with what we learned from experimental economics with human subjects [96].

3.2 Competition

In addition to mirroring the collective behavior of human agents, software agents are also used directly to interact with human agents. This advancement is partially

[18] The number of bargaining strategies assigned to each bargaining agent is called the *population size*. **AIE-DA Version 2**, developed by the AI-ECON Research Center, allows each agent to have at most 1000 bargaining strategies.

motivated by the increasing popularity of electronic commerce. In web-based online markets, such as *ebay*, one general question concerns the role of software agents. Can software agents perform better than human agents in making deals? This question is similar to the inquiry on the degree of smartness of software agents in comparison with that of human agents, which is a generalization of the issue pursued in Section 2.3.1. Of course, it would be problematic to measure the IQ of software agents. Nevertheless, the Turing test does challenge the possibility that one can measure the IQ of software agents if they are properly designed. As a result, as an extension of Section 2.3.2, we can now match human agents with software agents in various kinds of markets or games.

The agent-based financial system **U-MART** provides one of the best illustrations. U-MART stands for *UnReal Market as an Artificial Research Test bed*. This is a research project collaboratively initiated by a number of universities in Japan [95]. It is an agent-based future market, which serves both purposes of education and research. Second, U-MART enables us to address a very basic question: can human agents dominate the software agents when they are placed together in the market? What was found in some limited experiments is that the performance of software agents was superior to that of human agents. Of course, the result is still premature. Even the question is not well-defined. This is so because the competition was generally not made on a fair basis. For example, during the transaction process, human agents were poorly equipped with computational facilities so that they were not able to figure out some important figures in the nick of time. The competition will be considered more fair if it allows human agents to write their own trading programs and modify the program on-line, as will be discussed in Section 3.3.

Other interesting research questions can also be addressed by watching the interaction between human and software agents. Market efficiency is a case in point. [53] studied whether market efficiency can be enhanced when software agents are introduced to the markets which were originally composed solely of human agents.[19] 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. They then went further to distinguish the case where human agents are informed of the presence of software agents from the case where human agents are not informed of this presence. 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.[20] In a sense, this question can be viewed in terms of the *socio-psychological impact* on human behavior in the presence of interacting machine intelligence.

[19] See also [91] and [101].

[20] These issues have been further pursued in the recent development of the **U-Mart** platform ([99], [92], [65]).

3.3 Collaboration

At the third stage, neither do we mirror, nor do we match, the two kinds of agents. Human agents now work with software agents as a *team*, and they are no longer treated as two entities. The modern definition of artificial intelligence has already given up the dream of passing the Turing test. Instead, a more realistic and also interesting definition is based on the team work cooperatively performed by software agents and human agents [68]. If the studies of the first two stages can be considered to be the works under the influence of classical AI, then the third development is a natural consequence of the modern AI.

3.3.1 Is Collaboration Behaviorally Feasible?

This stage of research involves one non-trivial question: *is collaboration behaviorally feasible?* Would human agents choose to work with software agents if they were given this option? Can we see the collaborative computational intelligence beyond software agents in this way? [28] carried out an experiment to make software agents available in a laboratory with human subjects, and to watch whether human agents will choose software agents to be their representatives, while forming their decisions and actions. The laboratory used to facilitate this study is a *prediction market*, which is a kind of web-based and agent-based simulation platform [102]. They designed the software agents as the trading algorithms which can execute bid orders or ask orders when the market timing condition is met. Human agents have the option of choosing these software agents, and even rewrite these algorithms, to be their trading representatives. If they do so, their identities are completely characterized by the associated software agents. In this case, the distinction between human and software agents becomes problematic.

They carried out the above experiment in relation to a recent political event (a mayoral election in Taiwan), and found that out of 425 market participants (human agents), only 25 used software agents to trade. In other words, most human agents chose not to bother with software agents. This result is evidence that human agents may not collaborate with software agents. They then went further to ask the rationale behind the choice to ignore software agents. They first found that human agents who used software agents to trade generally earned higher profits than those who did not use them, and this dominance was statistically significant. This brought us face to face with a puzzling question: Why do human agents not "recruit" software agents when they are available in a promising way? To tackle this question, they distributed questionnaires to market participants to gain more information regarding their choice behavior.

3.3.2 Collaborate with Customized Software Agents

One of the reasons why human agents did not use software agents as their representatives was that they did not feel comfortable with them, or else they did not quite trust them. In other words, these software agents were not customized. In a subsequent experiment, they considered a different setup which may blur the relations between human and software agents. That is, they asked each human agent to write his or her own trading program (software agents), and used them as their incarnations in the later agent-based market simulation. This idea is very similar to the game-like tournaments pioneered by Robert Axelrod in the mid-1980s [12, 63], and the market-like tournament initiated by the Santa Fe Institute in the early 1990s [86]. However, moving one step further, they considered a comparison between the simulation based on these humanly-supplied software agents and the one based on purely computer-generated software agents. For the latter case, genetic programming was applied to generate the autonomous agents, and the platform **AIE-DA** was used to implement the simulation (see also Section 3.1.2).

Out of the 20 simulations which they carried out, they found that the market composed of purely computer-generated software agents that are autonomous and adaptive performs consistently better, in terms of market efficiency, than the market composed of humanly-supplied software agents, even though humanly-supplied software agents are more sophisticated or thoughtful.

The two experiments above together have two implications. The first experiment, from a sociological viewpoint, provides evidence that human agents may have difficulty embracing (containing) software agents when making decisions. The second experiment further shows that if we allow human agents to choose or even design their own software agents, their collective behavior may not be the same as that observed from the computer-generated software agents. This second finding is similar to that of [8].[21]

4 Hybrid Systems

Section 2 reviews the applications of CCI to *agent-based computational economics* (the heterogeneous-agent research paradigm), and Section 3 reviews the applications of CCI to *experimental economics*. In both of these two cases, CCI is mainly used to build software agents in economic and financial models. In other words, CCI contributes to *economic and financial agent engineering*. Another major area to which CCI is also vastly applied, with an even longer history, is *financial engineering*. This application is mainly motivated by the rapid development of various hybridizations of CI tools. There are a number of hybrid systems frequently observed in financial engineering. We shall only sample some in this section.

4.1 Nature of Hybridization

The idea of using various CI tools as a *module* of a hybrid system or an *agent* of a multi-agent system is not new, but it has just gained its momentum over the last

[21] [8] find significant differences in data generated by some chosen learning models and humans, with the greatest ones in coordination games.

decade. [22] The room for this collaboration is available mainly because of the heterogeneity of CI tools (see also Section 2). Different CI tools are designed with different mechanisms inspired from various natural and artificial processes; therefore, they may each handle one or a few different aspects of an intelligent task [23]. To name a few, self-organizing maps are operated for pattern discovery and concept formation, auto-association neural networks are good for the removal of redundancies and data compression, feedforward and recurrent neural networks are regarded as universal function approximators, and the approximation process can usually be facilitated by genetic algorithms. With this diversity in specialization, it would be surprising to see very little collaboration but much competition among them, as has been developed in the literature over the past. The recent research trend seems to recognize this biased development and move back to the collaboration theme accordingly.

Financial hybrid systems are mainly the financial applications of the hybrid systems or multi-agent systems. Among the many designs of financial hybrid systems, it is important to distinguish *models* from *processes*, particularly, *evolutionary processes*. Many financial hybrid systems are designed based on the idea of putting a model or a population of models under an evolutionary process, which includes evolutionary artificial neural networks, evolutionary fuzzy inference systems, evolutionary support vector machines, etc. We shall start with a review of this main idea (Section 4.2). The second major element we experienced in hybrid financial systems is the desire to make semantic sense of the hybrid systems, which includes the use of natural language and qualitative (non-numeric) reasoning. We shall then provide a brief review of this (Section 4.3). We conclude this section by mentioning the collaboration work done with the data or database, such as feature selection, dimension reduction, etc. (Section 4.4).

4.2 Evolutionary-Based Hybridization

The idea of the evolutionary-based hybridization is clear: there are two major elements in this hybrid system. One is the *universal function approximator*, and the other is the *built-in approximation process*. If we consider the former as an parametric model, then the latter can be regarded as an estimator of it. In economics, there are three frequently used evolutionary-based hybrid systems, namely, evolutionary artificial neural networks (Section 4.2.1), evolutionary support vector machines (Section 4.2.2), and evolutionary fuzzy or neurofuzzy inference systems.

4.2.1 Evolutionary Artificial Neural Networks

Among all hybridizations of CI in finance, the most popular one is probably the combination of genetic algorithms and artificial neural nets, which is one of the kinds of *evolutionary artificial neural nets* (EANNs), referred as to GANNs (genetic

[22] A more comprehensive treatment of the hybrid system can be found in [18] and [41].

algorithms + neural nets). EANNs are computationally very demanding; therefore, despite their conceptual appeal, there were few financial applications using EANNs in the 1990s. However, recently, a number of commercial algorithms, such as *NeuroSolutions*, have been made available, and hence the adoption of EANNs has spread quickly and widely.

In GANNs, genetic algorithms are applied to evolve an artificial neural net so as to genetically determine not only its weights, but also its topology, including the number of layers, the number of hidden nodes and input selection. For example, [64] uses a genetic algorithm to simultaneously optimize the connection weights between layers and select instance. It is found that genetically selected instances shorten the learning time and enhance prediction performance. It many applications, the weight determination is performed through the backpropagation algorithm, but in this case genetic algorithms can be applied to determine the learning rate and momentum of the backpropagation algorithm [94].

In addition to GAs, another subclass of EANNs is GPNNs (genetic programming + neural nets). [3] employ genetic programming to evolve artificial neural networks, and the genetically evolved neural networks are applied to forecast exchange rate returns for the Japanese Yen and the British Pound against the US dollar. The empirical results show the existence of a short-term weak predictable structure for both currencies.

Not being confined to feedforward neural nets, GAs have also been used to evolve other kinds of neural nets. [94] use a GA to genetically evolve recurrent neural networks. In this study, a hybrid model that combines a seasonal linear time series model and a recurrent neural network is used to forecast agricultural commodity prices. A genetic algorithm is employed to determine the optimal architecture of the ANNs. It turns out that the out-of-sample prediction can be improved slightly with the hybrid model. [105] uses a GA to evolve *fuzzy neural networks* for financial forecasting. In this study, the genetic algorithm and the gradient decent (GD) algorithm are used alternatively in an iterative manner to optimize all parameters and weights in a five-layer fuzzy neural network. It is found that the hybrid learning algorithm combining GA and GD is more powerful than the previous separate sequential trading algorithm.

The most active financial domain using EANNs is financial time series forecasting. [83] apply genetically evolved neural network models to predict the Straits Times Index (STI) of the Stock Exchange of Singapore. This study shows that satisfactory results can be achieved when applying genetically evolved neural networks to predict the STI. EANNs' applications to other financial domains include *financial distress prediction* [78] and *trading* [93, 14]. [93] enhance EANNs with fractal analyses. Based on Hurst exponent calculations, the appropriate input windows for the EANN are identified. It then investigates the efficacy of the model using closing price time series for a suite of stocks listed on the SPI index on the Australian Stock Exchange. The results show that Hurst exponent configured models out-perform basic EANN models in terms of average trading profit found using a simple trading strategy.

4.2.2 Evolutionary Support Vector Machines

In the 1990s, based on results from *statistical learning theory*, an alternative to the artificial neural network was developed, i.e. the support vector machine (SVM), also called the *kernel machine*. It has been found that when compared with the standard feedforward neural nets trained with the backpropagation algorithm, support vector machine can have superiorperformance [37]. This superiority may be attributable to different optimization principles running behind the two: for the SVM, it is the *structural risk minimization principle*, whereas for backpropagation it is the *empirical risk minimization principle*. The objective of the former is to minimize the upper bound of the generalization error, whereas the objective of the latter is to minimize the error based on training data. Hence, the former may lead to better generalization than the latter. Partially due to this advantage, the financial applications of SVM have kept on expanding.[23]

However, like the ANN, the SVM can also be treated as a semi-parametric model. To use it, there are a number of parameters or specifications that need to be determined. Basically, there are three major parameters in the SVM. First, there is the penalty parameter associated with the empirical risk appearing in the structural risk function. In the literature, it is normally denoted by C. Second, when the SVM is applied to the regression problem, in addition to C, there is a parameter ε appearing in the ε-error intensive function. Third, it is the parameter of the chosen kernel function. Support vector machines non-linearly map a lower dimensional input space into a high dimensional, possibly, an infinite dimensional, feature space. However, 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. There are two kernel functions frequently used, namely, the *Gaussian kernel* and the *polynomial kernel*. The former has a parameter associated with the second moment of a Gaussian called *width* (normally denoted by σ), and the latter has a parameter associated with the polynomial degree (normally denoted by p).

At the beginning, these parameters were arbitrarily given by trial-and-error. Later on, genetic algorithms were extensively employed to optimize the SVM, and the applications of ESVM have been seen in various fields, such as reliability forecasting [81], traffic flow forecasting [97], bankruptcy forecasting [1, 103, 106], and stock market prediction [107, 40].

[103] uses a GA to genetically determine the parameters C and σ of the SVM. The proposed GA-SVM was then tested on the prediction of financial crisis in Taiwan. The experimental results show that the GA-SVM model performs better than the manually-driven SVM. [1] use a GA to simultaneously optimize the feature selection and the instance selection as well as the parameters of a kernel function. It is also found in this study that the prediction accuracy of the conventional SVM may be improved significantly by using the ESVM. In [107] a GA is used for variable selection in order to reduce the modeling complexity of the SVM and improve the

[23] The interested reader can find some useful references directly from the website of the SVM: http://www.svms.org/

speed of the SVM, and then the SVM is used to identify the stock market movement direction based on historical data.

4.3 Semantics-Based Hybrid Systems

Fuzzy-based modeling is appealing for social scientists because the use of linguistic variables enables them to make semantic sense of their models, which generally leads to easy interpretation of the rules or knowledge extracted from the models. By contrast, many so-called “black-box” CI tools are often criticized for the lack of this property. Therefore, fuzziness becomes an imperative element for building intelligent systems, and, like an evolutionary mechanism, its necessity generates another series of hybrid systems, that we generally call *semantics-based hybrid systems*. From our point of view easy to understand is the distinguishing feature of semantics-based hybrid systems, and among all economic and financial applications of them, *neuro-fuzzy systems* are probably the most popular ones.

A neuro-fuzzy system is a fuzzy system that is represented by a kind of neural network, for example, the feedforward neural network, and, therefore, it can be trained (estimated) by using the associated learning algorithms of the network, e.g., backpropagation. Therefore, the neuro-fuzzy system is a *learning fuzzy system* in which the neural network learning algorithms are used to determine parameters of the fuzzy system. Several different neuro-fuzzy systems exist in the literature [80], but only the *ANFIS* (Adaptive Network-based Fuzzy Inference System) is widely used in economic and financial applications.

4.3.1 ANFIS

The ANFIS was proposed by [59]. It represents a *Sugeno-style* fuzzy system in a special five-layer feedforward neural network. The Sugeno style of a fuzzy if-then rule is:

If x is “A” and “y” is “B”, then $z = f(x,y)$.

“A” and “B” above are fuzzy sets. However, the function $f(x,y)$ in the ANFIS is linear:

$$z = f(x,y) = \alpha + \beta_x x + \beta_y y. \tag{26}$$

The structure can, therefore, be comparable to the autonomous-agent design in the SFI artificial stock market (see Section 2.2.3, Equation 13) and is even closer to the modified version of the SFI-ASM proposed by [98] (see Section 2.4). However, unlike [98], the rule base used in the ANFIS must be known in advance. The ANFIS integrates the backpropagation algorithm with the recursive least squares algorithm to adjust parameters.

The ANFIS has been applied to water consumption forecasting [11], stock prices forecasting [13, 39, 108], credit scoring [73], market timing decisions [38], credit risk evaluation [104], and option pricing [62].

[62] applies the ANFIS to option market pricing based on the transaction data of the Indian Stock Option. The pricing capability of the ANFIS is compared with the performance of the ANN model and Black-Scholes (BS) model. The empirical results show that the out-of-sample pricing performance of the ANFIS is superior to that of the BS, and is also better than the ANN. In addition, compared to the ANN, the ANFIS is explicit about its decision rules.

Instead of backpropagation, [39] uses *extended Kalman filtering* to estimate the ANFIS, and demonstrates its performance by comparing it with the ANFIS with regard to stock index forecasting. It is found that the proposed extended Kalman filtering can perform better than backpropagation.

4.3.2 Other Neuro-fuzzy Systems

In addition to the ANFIS, financial applications using other neuro-fuzzy systems also exist. [46] present a cooperative neuro-fuzzy inference system to forecast the expected financial performance of farm businesses. The fuzzy inference system considered is the *Mamdani Style* rather than the Sugeno Style generally used in the ANFIS. In addition, [46] only estimates the rule weight, and the parameters of the membership function, which is a sigmoid function, are not part of it. Therefore, it uses Kosko's FAM (*fuzzy associative memories*) vector quantization algorithm (or competitive learning algorithm) [66] to estimate the fuzzy inference system. The proposed system is compared with the conventional ordered multinomial logit regression model. The result shows that logit regression generally classified farms more accurately, but the FAM model was more accurate at predicting poorly performing farms, and, more importantly, the development and interpretation of the NFIS was found to be very intuitive.

[100] propose a new neural fuzzy system, namely the *generic self-organizing fuzzy neural network* based on the *compositional rule of inference*, as an alternative to predict banking failure. The system referred to as GenSo is able to identify the inherent traits of financial distress based on financial features derived from publicly available financial statements. The interaction between the selected features is captured in the form of highly intuitive if-then fuzzy rules. Such rules hence provide insights into the possible characteristics of financial distress and form the knowledge base for a highly desired early warning system that aids bank regulation.

4.4 Feature Reduction: Rough GA or GP

The hybrid system which we shall review in this section comprises two CI tools, namely, genetic programming and rough sets. The hybridization of GP and rough sets provides an excellent illustration of how the usual competitive relationship between two CI tools can be more productively transformed into into a collaborative relationship ([77, 88]).

Rough sets define a mathematical model of vague concepts that is used to represent the relationship of dissimilarity between objects. Two objects are considered *equivalent* with respect to *a certain subset of attributes* if they share the same value for each attribute of the subset. By collecting all equivalent objects, one can decompose the entire universal (set of objects) into equivalent classes. The decomposition, of course, is not unique and is dependent on the subset of attributes which we use to define the equivalent relation.

Rough sets arise when one tries to use the equivalent classes with respect to some attributes to give a description of a concept based on the associated decision attributes. For example, if the decision attribute concerns financial distress and is binary (bankruptcy or not), then what one wants to characterize is the concept of *bankruptcy* by using some attribute of the firms, e.g., their financial ratios, size, etc. The characterizations are only approximate when complete specification of the concept is infeasible. In this case, the concept itself is rough, and the objects associated with the concept are referred to as the rough set.

Two partial specifications are considered to be the most important, namely, the *lower approximations* and the *upper approximations*. The lower approximations consist of objects (equivalent classes) which belong to a concept with certainty, i.e., the entire equivalent classes are a subset of the rough set. The upper approximations consist of those equivalent classes which possibly belong, i.e., they have non-empty intersection with the rough set. A subset of the attributes is called *reduct* if all attributes belonging to it are *indispensable*. An attribute is *dispensable* if its absence does not change the set approximation. In other words, a reduct is a set of attributes that is sufficient to describe a concept.

A financial hybrid system using rough sets and GP is first proposed by [77]. In this hybridization, the rough set is firstly adopted to select the discriminative features by identifying *reducts*. Only these reducts are then taken as the input features for the GP learning process. [77] uses genetic programming to construct a bankruptcy prediction model with variables from a rough sets model. The genetic programming model reveals relationships between variables that are not apparent in using the rough sets model alone.[24]

5 Concluding Remarks

According to the current trend in the literature, this paper addresses what collaborative computational intelligence can mean for economists. While the recent series of publications on the economic and financial applications of computational intelligence has already demonstrated the relevance of various CI tools to economists [29, 35], they are mostly taken as techniques for economists. In this chapter, we go

[24] There are many other ways to hybridize GP or GA with rough sets, but so far their financial applications have rarely been seen.

one step further to show that they can be more productive so as to be part of the future of economics. Specifically, we demonstrate this potential by singling out two new research paradigms in economics, namely, agent-based economics and experimental economics.

The essence of agent-based economics is a society of heterogeneous agents, a subject which is highly interdisciplinary motivated. Collaborative computation intelligence enables or inspires economists to see how some initial explorations of the richness of this society can be made. In this regard, computational intelligence may contribute by providing not just models of learning or adaptation, but models of learning or adaptation processes which may be influenced by behavioral genetic and cultural factors.

After one decade of rapid development, a challenging issue facing experimental economics is how to strengthen the reliability of the laboratory results with human subjects by properly introducing software agents to labs. In fact, the state-of-art economic laboratory is no longer a lab with only human subjects, but a lab comprising both human agents and software agents [27]. Collaborative computational intelligence can contribute significantly to the building of the modern lab.

The last part of the paper reviews some recent economic and financial applications of hybrid systems. However, there is no attempt to give an exhaustive list, which itself may deserve a separate treatment. We, therefore, single out the two most significant elements in frequently used economic and financial hybrid systems, namely, evolution and semantics. The former mainly contributes to the hybrid system as a process to facilitate the universal approximation, whereas the latter contributes to the hybrid system by enhancing its semantics.

To sum up, this chapter has shown how collaborative computational intelligence has enriched the design of economic and financial agents, while, in the meantime, providing quantitative economists with a longer list of ideas to cope with the inherent complexity and uncertainty in the data.

Acknowledgements. The author is grateful to one anonymous referee for his helpful suggestion. The author is also grateful to the editor, Dr. Christine L. Mumford, for her devoted efforts made in editing the chapter. NSC research grant No. 95-2415-H-004-002-MY3 is gratefully acknowledged.

References

1. Ahn, H., Lee, K., Kim, K.-J.: Global optimization of support vector machines using genetic algorithms for bankruptcy prediction. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) ICONIP 2006. LNCS, vol. 4234, pp. 420–429. Springer, Heidelberg (2006)
2. Allen, H., Taylor, M.: Charts, noise and fundamentals in the London foreign exchange market. Economic Journal 100, 49–59 (1990)
3. Alvarez-Diaz, M., Alvarez, A.: Forecasting exchange rates using an evolutionary neural network. Applied Financial Economics Letters 3(1), 5–9 (2007)
4. Amilon, H.: Estimation of an adaptive stock market model with heterogeneous agents. Journal of Empirical Finance (forthcoming) (2008)

5. Aoki, M., Yoshikawa, H.: Reconstructing macroeconomics: A perspective from statistical physics and combinatorial stochastic processes. Cambridge University Press, Cambridge (2006)
6. Arifovic, J.: Genetic algorithm learning and the cobweb model. Journal of Economic Dynamics and Control 18(1), 3–28 (1994)
7. Arifovic, J.: Genetic algorithms and inflationary economies. Journal of Monetary Economics 36(1), 219–243 (1995)
8. Arifovic, J., McKelvey, R., Pevnitskaya, S.: An initial implementation of the Turing tournament to learning in repeated two-person games. Games and Economic Behavior 57(1), 93–122 (2006)
9. Arthur, W.: Inductive reasoning and bounded rationality. In: American Economic Association Papers Proceedings, vol. 84, pp. 406–411 (1994)
10. Arthur, W., Holland, J., LeBaron, B., Palmer, R., Tayler, P.: Asset pricing under endogenous expectations in an artificial stock market. In: Arthur, B., Durlauf, S., Lane, D. (eds.) The economy as an evolving complex system II, pp. 15–44. Addison-Wesley, Reading (1997)
11. Atsalakis, G., Ucenic, C.: Time series prediction of water consumption using the neuro-fuzzy (ANFIS) approach. In: IWA International Conference on Water Economics, Statistics, and Finance, Rethymno, Greece, July 8–10, vol. I, pp. 93–100 (2005)
12. Axelrod, R.: The evolution of cooperation. Basic Books (1984)
13. Azeem, M., Hanmandlu, M., Ahmad, N.: Generalization of adaptive neuro-fuzzy inference systems. IEEE Transactions on Neural Networks 11(6), 1332–1346 (2000)
14. Azzini, A., Tettamanzi, A.: Evolving neural networks for static single-position. Journal of Artificial Evolution and Applications 2008, 1–17 (2008) Article ID 184286
15. Barber, B., Odean, T.: Trading is hazardous to your wealth: The common stock investment performance of individual investors. Journal of Finance 55, 773–806 (2000)
16. Brabazon, A., O'Neill, M. (eds.): Natural computing in computational economics and finance. Springer, Heidelberg (2008)
17. Brenner, T.: Agent learning representation: Advice on modeling economic learning. In: Tesfatsion, L., Judd, K. (eds.) Handbook of computational economics: Agent-based computational economics, vol. 2, pp. 895–947. Elsevier, Oxford (2006)
18. Bonissone, P.: Hybrid soft computing systems: Where are we going? In: Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000), Berlin, Germany, pp. 739–746 (2000)
19. Carlin, B., Louis, T.: Bayes and empirical Bayes methods for data analysis, 2nd edn. CRC Press, Boca Raton (2000)
20. Casella, G.: An introduction to empirical Bayes data analysis. The American Statistician 39, 83–87 (1985)
21. Chan, N., LeBaron, B., Lo, A., Poggio, T.: Information dissemination and aggregation in asset markets with simple intelligent traders, Working paper. MIT, Cambridge (1999)
22. Chen, S.-H.: Software-agent designs in economics: An interdisciplinary framework. IEEE Computational Intelligence Magazine 3(4), 22–26 (2008a)
23. Chen, S.-H.: Computational intelligence in agent-based computational economics. In: Fulcher, J., Jain, L. (eds.) Computational intelligence: A compendium, pp. 517–594. Springer, Heidelberg (2008)
24. Chen, S.-H., Kuo, T.-W.: Overfitting or poor learning: A critique of current financial applications of GP. In: Ryan, C., Soule, T., Keijzer, M., Tsang, E.P.K., Poli, R., Costa, E. (eds.) EuroGP 2003. LNCS, vol. 2610, pp. 34–46. Springer, Heidelberg (2003)
25. Chen, S.-H., Huang, Y.-C.: Risk preference, forecasting accuracy and survival dynamics: Simulations based on a multi-asset agent-based artificial stock market. Journal of Economic Behavior and Organization (forthcoming) (2008)

26. Chen, S.-H., Tai, C.-C.: Trading restrictions, price dynamics, and allocative efficiency in double auction markets: Analysis based on agent-based modeling and simulations. Advances in Complex Systems 6(3), 283–302 (2003)
27. Chen, S.-H., Tai, C.-C.: On the selection of adaptive algorithms in ABM: A computational-equivalence approach. Computational Economics 28(1), 51–69 (2006)
28. Chen, S.-H., Tai, C.-C.: Would human agents like software agents? Results from prediction market experiments. Working paper, AI-ECON Research Center, National Chengchi University (2007)
29. Chen, S.-H., Wang, P.: Computational intelligence in economics and finance. Springer, Heidelberg (2003)
30. Chen, S.-H., Yeh, C.-H.: Genetic programming learning and the cobweb model. In: Angeline, P. (ed.) Advances in Genetic Programming, vol. 2, ch. 22, pp. 443–466. MIT Press, Cambridge (1996)
31. Chen, S.-H., Yeh, C.-H.: Modeling speculators with genetic programming. In: Angeline, P.J., McDonnell, J.R., Reynolds, R.G., Eberhart, R. (eds.) EP 1997. LNCS, vol. 1213, pp. 137–147. Springer, Heidelberg (1997)
32. Chen, S.-H., Yeh, C.-H.: Simulating economic transition processes by genetic programming. Annals of Operation Research 97, 265–286 (2000)
33. Chen, S.-H., Yeh, C.-H.: Evolving traders and the business school with genetic programming: A new architecture of the agent-based artificial stock market. Journal of Economic Dynamics and Control 25, 363–394 (2001); Chen, S.-H., Kuo, T.-W., Shieh, Y.-P.: Genetic Programming: A Tutorial with the software Simple GP. In: Chen, S.-H. (ed.) Genetic algorithms and genetic programming in computational finance, pp. 55–77. Kluwer, Dordrecht (2002)
34. Chen, S.-H., Liao, C.-C., Chou, P.-J.: On the plausibility of sunspot equilibria: Simulations based on agent-based artificial stock markets. Journal of Economic Interaction and Coordination 3(1), 25–41 (2008)
35. Chen, S.-H., Wang, P., Kuo, T.-W.: Computational intelligence in economics and finance, vol. 2. Springer, Heidelberg (2007)
36. Chen, S.-H., Zeng, R.-J., Yu, T.: Co-evolving trading strategies to analyze bounded rationality in double auction markets. In: Riolo, R. (ed.) Genetic programming theory and practice VI. Springer, Heidelberg (forthcoming) (2008)
37. Chen, W.-H., Shih, J.-Y., Wu, S.: Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets. International Journal of Electronic Finance 1(1), 49–67 (2006)
38. Chen, P., Quek, C., Mah, M.: Predicting the impact of anticipatory action on U.S. stock market– An event study using ANFIS (a neural fuzzy model). Computational Intelligence 23(2), 117–141 (2007)
39. Chokri, S.: Neuro-fuzzy network based on extended Kalman filtering for financial time series. In: Proceedings of World Academy of Science, Engineering and Technology, vol. 15, pp. 290–295 (2006)
40. Choudhry, R., Garg, K.: A hybrid machine learning system for stock market forecasting. In: Proceedings of the World Academy of Science, Engineering and Technology, vol. 29, pp. 315–318 (2008)
41. Cordon, O., Herrera, F., Hoffmann, F., Magdalena, L.: Genetic fuzzy systems: Evolutionary tuning and learning of fuzzy knowledge bases. World Scientific, Singapore (2002)
42. De Grauwe, P., Grimaldi, M.: The exchange rate in a behavior finance framework. Princeton University Press, Princeton (2006)

43. de Jong, E., Verschoor, W., Zwinkels, R.: Heterogeneity of agents and exchange rate dynamics: Evidence from the EMS, http://ssrn.com/abstract=890500
44. Delli Gatti, D., Gaffeo, E., Gallegati, M., Giulioni, G.: Emergent macroeconomics: An agent-based approach to business fluctuations. Springer, Heidelberg (2008)
45. Duffy, J.: Agent-based models and human subject experiments. In: Tesfatsion, L., Judd, K. (eds.) Handbook of computational economics: Agent-based computational economics, vol. 2, pp. 949–1011. Elsevier, Oxford (2006)
46. Duval, Y., Kastens, T., Featherstone, A.: Financial classification of farm businesses using fuzzy systems. In: 2002 AAEA Meetings Long Beach, California (2002)
47. Evans, G., Honkapohja, S.: Learning and expectations in macroeconomics. Princeton University Press, Princeton (2001)
48. Feinberg, M.: Why smart people do dumb things: Lessons from the new science of behavioral economics, Fireside (1995)
49. Fogel, D., Chellapilla, K., Angeline, P.: Evolutionary computation and economic models: sensitivity and unintended consequences. In: Chen, S.-H. (ed.) Evolutionary computation in economics and finance, pp. 245–269. Springer, Heidelberg (2002)
50. Frankel, J., Froot, K.: Chartists, fundamentalists, and trading in the foreign exchange market. American Economic Review 80, 181–186 (1990)
51. Gilks, W., Richardson, S., Spiegelhalter, D.: Markov chain Monte Carlo in practice. CRC, Boca Raton (1995)
52. Gode, D., Sunder, S.: Allocative efficiency of markets with zero intelligence traders: Market as a partial substitute for individual rationality. Journal of Political Economy 101, 119–137 (1993)
53. Grossklags, J., Schmidt, C.: Software agents and market (in)efficiency: a human trader experiment. IEEE Transactions on Systems, Man, and Cybernetics, Part C 36(1), 56–67 (2006)
54. Hartley, J.: The representative agent in macroeconomics. Routledge, London (1997)
55. Henrich, J.: Does culture Matter in Economic Behavior? Ultimatum game bargaining among the Machiguenga of the Peruvian Amazon. American Economic Review 90(4), 973–979 (2000)
56. Henrich, J., Boyd, R., Bowles, S., Camerer, C., Fehr, E., Gintis, H. (eds.): Foundations of human sociality: Economic experiments and ethnographic evidence from fifteen small-scale societies. Oxford University Press, Oxford (2004)
57. Herrnstein, R., Murray, C.: Bell curve: Intelligence and class structure in American life. Free Press (1996)
58. Hommes, C.: Heterogeneous agent models in economics and finance. In: Tesfatsion, L., Kenneth, J. (eds.) Handbook of Computational Economics, vol. 2, ch. 23, pp. 1109–1186. Elsevier, Amsterdam (2006)
59. Jang, R.: ANFIS: Adaptive network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics 23(3), 665–685 (1993)
60. Janssen, M., Ostrom, E.: Empirically based, agent-based models. Ecology and Society 11(2), 37 (2006)
61. Jensen, A.: The g Factor: The science of mental ability, Praeger (1998)
62. Kakati, M.: Option pricing using the adaptive neuro-fuzzy system (ANFIS). ICFAI Journal of Derivatives Markets 5(2), 53–62 (2008)
63. Kendall, G., Yao, X., Chong, S.-Y.: The iterated prisoners' dilemma: 20 years on. World Scientific, Singapore (2007)
64. Kim, K.-J.: Artificial neural networks with evolutionary instance selection for financial forecasting. Expert Systems with Applications 30(3), 519–526 (2006)

65. Koyama, Y., Sato, H., Matsui, H., Nakajima, Y.: 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., Deguchi, H. (eds.) Agent-based simulation: From modeling methodologies to real-world applications. Springer Series on Agent-Based Social Systems, vol. 1, pp. 158–166 (2005)
66. Kosko, B.: Neural networks and fuzzy systems. Prentice-Hall, Englewood Cliffs (1992)
67. LeBaron, B.: Evolution and time horizons in an agent based stock market. Macroeconomic Dynamics 5, 225–254 (2001)
68. Lieberman, H.: Software agents: The MIT approach. In: Invited speech delivered at the 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW 1996), Eindhoven, The Netherlands, January 22-25 (1996)
69. Lubinski, D., Humphreys, L.: Incorporating general intelligence into epidemiology and the social sciences. Intelligence 24(1), 159–201 (1997)
70. Lucas Jr., R.: Adaptive behavior and economic theory. Journal of Business 59, 401–426 (1986)
71. Lynn, R.: Race differences in intelligence: An evolutionary analysis. Washington Summit Publishers (2006)
72. Lynn, R., Vanhanen, T.: IQ and the wealth of nations. Praeger (2002)
73. Malhotra, R., Malhotra, D.: Differentiating between good credits and bad credits using neuro-fuzzy systems. European Journal of Operational Research 136(1), 190–211 (2002)
74. Markose, S.: Developments in experimental and agent-based computational economics (ACE): overview. Journal of Economic Interaction and Coordination 1(2), 119–127 (2006)
75. Manzan, S., Westerhoff, F.: Heterogeneous expectations, exchange rate dynamics and predictability. Journal of Economic Behavior and Organization 64, 111–128 (2007)
76. McClearn, G., Johansson, B., Berg, S., Pedersen, N., Ahern, F., Petrill, S., Plomin, R.: Substantial genetic influence on cognitive abilities in twins 80 or more years old. Science 276, 1560–1563 (1997)
77. McKee, T., Lensberg, T.: Genetic programming and rough sets: A hybrid approach to bankruptcy classification. European Journal of Operational Research 138(2), 436–451 (2002)
78. Mora, A., Alfaro-Cid, E., Castillo, P., Merelo, J., Esparcia-Alcazar, A., Sharman, K.: Discovering causes of financial distress by combining evolutionary algorithms and artificial neural networks. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008), Atlanta, USA (July 2008)
79. Murray, C.: Income Inequality and IQ. AEI Press (1998)
80. Nauck, D., Klawonn, F., Kruse, R.: Foundations of neuro-fuzzy systems. John Wiley and Sons, Chichester (1997)
81. Pai, P.-F.: System reliability forecasting by support vector machines with genetic algorithms. Mathematical and Computer Modelling 43(3-4), 262–274 (2006)
82. Palmer, R., Arthur, B., Holland, J., LeBaron, B., Tayler, P.: Artificial economic life: A simple model of a stock market. Physica D 75, 264–274 (1994)
83. Phua, P., Ming, D., Li, W.: Neural network with genetically evolved algorithms for stocks prediction. Asia-Pacific Journal of Operational Research 18(1), 103–107 (2001)
84. Plomin, R., Petrill, S.: Genetics and intelligence: What's new? Intelligence 24(1), 53–77 (1997)
85. Richiardi, M., Leombruni, R., Contini, B.: Exploring a new ExpAce: The complementarities between experimental economics and agent-based computational economics. Journal of Social Complexity 3(1) (2006)

86. Rust, J., Miller, J., Palmer, R.: Behavior of trading automata in a computerized double auction market. In: Friedman, D., Rust, J. (eds.) The double auction market: Institutions, theories, and evidence, ch. 6, pp. 155–198. Addison Wesley, Reading (1993)
87. Rust, J., Miller, J., Palmer, R.: Characterizing effective trading strategies: Insights from a computerized double auction market. Journal of Economic Dynamics and Control 18, 61–96 (1994)
88. Salcedo-Sanz, S., Fernandez-Villacanas, J.-L., Segovia-Vargas, M., Bousono-Calzon, C.: Genetic programming for the prediction of insolvency in non-life insurance companies. Computers & Operations Research 32(4), 749–765 (2005)
89. Sargent, T.: Bounded rationality in macroeconomics. Oxford University Press, Oxford (1993)
90. Sasaki, Y., Flann, N., Box, P.: Multi-agent evolutionary game dyanmics and reinforcement learning applied to online optimization for the traffic policy. In: Chen, S.-H., Jain, L., Tai, C.-C. (eds.) Computational economics: A perspective from computational intelligence. IDEA Group Publishing, USA (2005)
91. Sato, H., Matsui, H., Ono, I., Kita, H., Terano, T., Deguchi, H., Shiozawa, Y.: U-Mart project: Learning economic principles from the bottom by both human and software agents. In: Terano, T., Nishida, T., Namatame, A., Tsumoto, S., Ohsawa, Y., Washio, T. (eds.) JSAI-WS 2001. LNCS, vol. 2253, pp. 121–131. Springer, Heidelberg (2001)
92. Sato, H., Kawachi, S., Namatame, A.: The statistical properties of price fluctuations by computer agents in a U-Mart virtual future market simulator. In: Terano, T., Deguchi, H., Takadama, K. (eds.) Meeting the challenge of social problems via agent-based simulation, pp. 67–76. Springer, Heidelberg (2003)
93. Selvaratnam, S., Kirley, M.: Predicting stock market time series using evolutionary artificial neural networks with Hurst exponent windows. In: Sattar, A., Kang, B.-h. (eds.) AI 2006. LNCS, vol. 4304, pp. 617–626. Springer, Heidelberg (2006)
94. Shahwan, T., Odening, M.: Forecasting agricultural commodity prices using hybrid neural networks. In: Chen, S.-H., Wang, P., Kuo, T.-W. (eds.) Computational intelligence in economics and finance, vol. 2, pp. 63–74. Springer, Heidelberg (2007)
95. Shiozawa, Y., Nakajima, Y., Matsui, H., Koyama, Y., Taniguchi, K., Hashimoto, F.: Artificial Market Experiments with the U-Mart. Springer, Tokyo (2006)
96. Smith, V.: Bidding and auctioning institutions: Experimental results. In: Smith, V. (ed.) Papers in eExperimental economics, pp. 106–127. Cambridge University Press, Cambridge (1991)
97. Su, H., Yu, S.: Hybrid GA based online support vector machine model for short-term traffic flow forecasting. In: Xu, M., Zhan, Y.-W., Cao, J., Liu, Y. (eds.) APPT 2007. LNCS, vol. 4847, pp. 743–752. Springer, Heidelberg (2007)
98. Tay, N., Linn, S.: Fuzzy inductive reasoning, expectation formation and the behavior of security prices. Journal of Economic Dynamics & Control 25, 321–361 (2001)
99. Terano, T., Shiozawa, Y., Deguchi, H., Kita, H., Matsui, H., Sato, H., Ono, I., Kakajima, Y.: U-Mart: An artificial market testbed for economics and multiagent systems. In: Terano, T., Deguchi, H., Takadama, K. (eds.) Meeting the challenge of social problems via agent-based simulation, pp. 53–66. Springer, Heidelberg (2003)
100. Tung, W.-L., Quek, C., Cheng, P.: GenSo-EWS: A novel neural-fuzzy based early warning system for predicting bank failures. Neural Networks 17(4), 567–587 (2003)
101. Ueda, T., Taniguchi, K., Nakajima, Y.: An analysis of U-Mart experiments by machine and human agents. In: Proceedings of 2003 IEEE international symposium on computational intelligence in robotics and automation, vol. 3, pp. 1340–1347 (2003)
102. Wang, S.-C., Li, S.-P., Tai, C.-C., Chen, S.-H.: Statistical properties of an experimental political futures markets. Quantitative Finance (2007) (forthcoming)

103. Wu, C.H., Tseng, G.H., Goo, Y.J., Fang, W.C.: A real-valued genetic algorithm to optimize the parameters of support vector machine for Predicting bankruptcy. Expert Systems with Applications 32(2), 397–408
104. Xiong, Z.-B., Li, R.-J.: Credit risk evaluation with fuzzy neural networks on listed corporations of China. In: Proceedings of 2005 IEEE International Workshop on VLSI Design and Video Technology, pp. 397–402 (2005)
105. Yu, L., Zhang, Y.-Q.: Evolutionary fuzzy neural networks for hybrid financial prediction. IEEE Transactions on Systems, Man and Cybernetics, Part C 35(2), 244–249 (2005)
106. Yu, L., Lai, K.-K., Wang, S.: An evolutionary programming based SVM ensemble model for corporate failure prediction. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4432, pp. 262–270. Springer, Heidelberg (2007)
107. Yu, L., Wang, S.-Y., Lai, K.K.: Mining stock market tendency using GA-based support vector machines. In: Deng, X., Ye, Y. (eds.) WINE 2005. LNCS, vol. 3828, pp. 336–345. Springer, Heidelberg (2005)
108. Yunos, Z., Shamsuddin, S., Sallehuddin, R.: Data modeling for Kuala Lumpur Composite Index with ANFIS. In: Proceedings of 2008 Second Asia International Conference on Modelling & Simulation, pp. 609–614 (2008)
109. Zopounidis, C., Doumpos, M., Pardalos, P. (eds.): Handbook of financial engineering. Springer, Heidelberg (2008)