Economic Models of Innovations: Why GP Can Be a Possible Way Out?

Shu-Heng Chen
AI-ECON Research Center
Department of Economics
National Chengchi University
Taipei, Taiwan 11623
chchen@nccu.edu.tw

Bin-Tzong Chie
AI-ECON Research Center
Department of Economics
National Chengchi University
Taipei, Taiwan 11623
chie@aiecon.org

Abstract

No matter how commonly the term *innovation* has been used in economics, a concrete analytical or computational model of innovation is not yet available. This paper argues that a breakthrough can be made with *genetic programming*, and proposes a functional-modularity approach to an agent-based computational economic model of innovation.

Motivation and Introduction

No matter how commonly the term "*innovation*" or "*technological progress*" has been used in economics, or more generally, in social sciences, a concrete analytical or computational model of innovation is not yet available. Studies addressing specific technology advancements in different scientific and engineering fields are, of course, not short; however, the *general representation* of technology, based on which innovation can be defined and its evolutionary process can be studied, does not exists.

For example, Figure 1 and 2 show the evolutionary processes of the hammer and the weapon. The general analytical or computational model which is able to demonstrate these evolutionary processes is not available. Kerber and Saam (2001) attempted to embed technology in a finite-dimensional space, and Ma and Nakamori (2002) used the Kauffman's famous KM model and genetic algorithms to represent technology and its evolution. We, however, argue that it is desirable to embed technology in an *infinite-dimensional space*. It would be inconceivable to think the evolution from the DOS system to the Windows system as just the fine-tuning of some parameters in a finite-dimensional space.

The lack of analytical or computational model of technology evolution is not surprising because *thinking how we think*, to a large extent, remains to be a daunting task for scientists. Therefore, in economics, a convenient device is simply to treat *technology* as a *parameter* in the model. For example, when coming to its effect on economic growth, technology is taken as a parameter in the production function,

$$Y = AF(L,K), \qquad (1)$$

where L and K refer to the input *labor* and *capital*, Y denotes the output, and A represents the technology level. In modern theory of growth, economists attempt to offer explanations for the advancement of technology by setting A as an output from another production function. Nevertheless, the inner structure of A has never been addressed in economics. If the inner structure is poorly understood, then it is hard to give any meaningful quantification of A. Consequently, so far, there is no direct measurement of A, and the best we have are only indirect measurements, such as the number of patents authorized, number of new publications,... Needless to say, none of these numbers helps us understand the nature and the structure of innovation processes.

While direct modeling of innovation is difficult, economists' dissatisfaction with neo-classical economic research paradigm is increasing, partially due to its incompetence to produce novelties (or the so-called *emergent property*). We cannot assume in advance that we know all new goods and new technology to be invented in the future. Therefore, in our model, we must leave a space to anticipate these unanticipated stuff. Recently, Aoki (2002a,b) introduced Zabell's notion of *unanticipated knowledge* to economists (Zabell, 1992). The notion is motivated by *population genetics*. In population genetics, unanticipated knowledge can be related to the sampling of species problem. In probability and statistics it is called the *law of succession*, i.e. how to specify the conditional probability that the next sample is never seen, given available sets of observation up to now. While Aoki's recent efforts indicate the fact that economists have not paid much attention to models of innovation, his proposed Ewens-Pitman-Zabell induction method is still rather limited. Basically, the nature of diversity of species and the nature of human creativity should not be treated equally (Basalla, 1988). Discovering new species is not a mentally creative activity: if a new species does not exist at all, we simply have no way to discover it. Furthermore, there is no reason why species discovery shall follow an evolutionary fashion, whereas the technological advancement is believed to have such continuous nature.

This paper, as the first stage of the research project "*Models of Innovation*" now launched at the *AI-ECON Research Center*, proposes *genetic programming* as a possible way leading to building an economic model of innovation. Our argument is based on two essential standpoints. First of all,

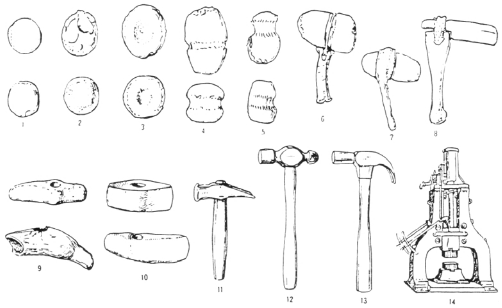

Figure 1: The Evolutionary History of the Hammer (Source: Basalla (1988), Figure 1.4, p.20.)

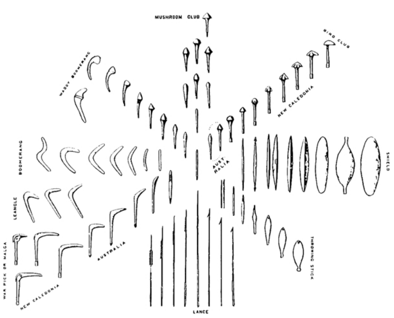

Figure 2: The Evolutionary History of the Weapon (Source: Basalla (1988), Figure 1.3, p.19.)

regarding the innovation process, we consider it as a *continuous process* (*evolution*), rather than a *discontinuous process* (*revolution*). By the continuous viewpoint, novel artifacts can only arise from antecedent artifacts. New kinds of made things are never pure creations. Basalla (1988) gave a vivid demonstration in pictures on the evolution of *weapons* and *hammers* (Figure 1 and 2). From this standpoint, it is clear that we are looking for an evolutionary model of innovation. Second, given the continuity argument, we assume that innovation is a *growing process*, i.e., combining low-level building blocks or features to achieve a certain kind of high-level functionality. In plain English, new ideas come from the use (the combination) of the old ideas (building blocks). New ideas, once invented, will become building blocks for other more advanced new ideas. In fact, the evolutionary process of the hammer and the weapon depicted by Basalla (1988) may be replicated by this functional-modularity approach.[1] That GP can deliver this feature has already been well evidenced on a series of its promising applications to the scientific, engineering, and financial domains.

[1]The idea of functional modularity is not new for economists. For example, Paul Romer already mentioned "Our physical world presents us with a relatively small number of building blocks–the elements of the periodic table–that can be arranged in an inconceivably large number of ways." (Romer, 1998)

Functional-Modularity Approach

To see the relevance of the functional-modularity approach to the evolutionary process of technology, it would be useful to start with an example. Figure 3 presents an application of the genetic algorithm to facial recognition. This application can be considered as an alternative to the traditional approach which relies on a witness to detail some features of the face he saw to a drawer. A group of faces are now automatically generated, and the witness only needs to input a score (fitness value) to each of the face based on its closeness to the true one he saw. Then, the genetic algorithm is applied to set an evolutionary process in motion to review and revise these faces. Hopefully, one of these faces coming out will eventually be very similar to the true one.

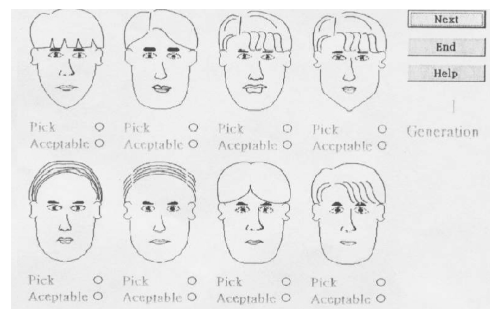

Figure 3: A Sample of Faces Random Initially Generated by the GA

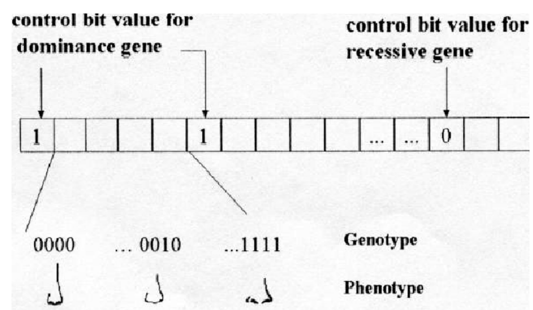

Figure 4: Binary Coding of a Face

Illustrations of GP in Discovering

A decade ago, financial economists already started to apply the functional-modularity approach with GP in discovering *new* trading rules. The issue is as follows: given some building blocks or knowledge from experts, can GP discover some profitable trading rules which are not known to us, including those experts? A series of experiments in the foreign exchange markets and the stock markets were carried out by (Neely ,Weller, and Dittmar, 1997; Neely and Weller, 1999; Allen and Karjalainen, 1999). They took *moving average rules* and *trading range break-out rules* as the building blocks (primitives). GP was employed to grow new trading rules from these primitives. They then tested the profitability of the rules discovered by GP, and examined the contents of these rules: *did the GP trading rules tell us anything more than just the simple technical trading rules?* In the foreign exchange markets, the result is promising. First, GP can discover profitable trading rules, and second, what GP discovered is actually much richer than what simple rules can tell us. Details of what was discovered by GP was also discussed in their studies. Hence, GP already demonstrated the innovation process of technical trading rules: combining low-level building blocks (MA, filter, or break-out rules) to achieve a certain kind of high-level functionality (profitable performance).

John Koza's application of GP to *Kepler's law* is another striking example. Here, not only did GP rediscover the law, but also, as the system climbed up the fitness scale, one of its interim solutions corresponds to an earlier conjecture by Kepler, published ten years before the great mathematician finally perfected the equation. (Levy, 1992; Banville, 1993) So, what GP presents is not just the end-result, but may replicate the whole trial-and-error process (the learning and probing process) which human may experience on their way to discovery. This is certainly a desirable feature to satisfy the continuity hypothesis of technology advancement. John Koza's another application of GP to analog circuits shows that GP-evolved solutions can actually compete with human ingenuity: the results closely matched ideas convinced by humans. Koza's GP has produced circuit designs that infringe on 21 patents in all, and duplicate the functionality of several others in novel ways (Willihnganz (1999)).

Elements

Commodities and Production

The illustrations above evidence that GP can simulate a functional-modularity continuous (evolutionary) innovation process. Nevertheless, applying GP to economic models of innovations, more precisely, an *agent-based computational economic model* of innovation, one shall anticipate a series of complications and difficulties which one does not encounter in just the applications to knowledge discovery and data mining. Let us point out a few of them in parlance of GP. First, *what would be the primitives or the initial building blocks?* This is a hard issue. Notice that we are only interested in a very *general* or *abstract* description of innovation, rather than any specific kind of innovation. This issue is hard because commodities in economic theory essentially has empty content. Little attention has been paid to its size, shape, topology, and inner structure. A general representation of commodities simply does not exists in current economic theory. As a result, the emerging process of new commodities from existing commodities, i.e., the result of technological innovation, has not been formally addressed in economic theory. This weak background offers us no guideline as to the choice of primitives.

In this paper, a breakthrough is made by first associating each commodity with its *production process*. Each production process is described by a sequence of processors and the materials employed. In general, each sequence may be furthered divided into many parallel subsequences. Different sequences (or subsequences) define different commodities. The commodity with the associated processor itself is also a processor whose output (i.e., the commodity) can be taken as a material used by an even higher level of production. With this structure, the function set naturally refers to a set of *primitive processors*, and the terminal set refers to a set of *raw materials*. They are denoted respectively as the following,

$$Terminal \quad Set : \Sigma = \{X_1, X_2, ..., X_\kappa\}, \tag{2}$$

$$Function \quad Set : \Xi = \{F_1, F_2, ..., F_k\}. \tag{3}$$

Each sequence (commodity, processor) can then be represented by a *LISP S-expression* or, simply, *a parse tree*.[2] The innovation process can then be simulated by the standard GP. Automatic defined functions (ADFs) are used to characterized some well-accepted processors developed during evolution. The function set is then adaptive with the deletion and addition of processors, including primitive functions and ADFs. The *knowledge* of the society at a point in time can then be measured by the complexity and the diversity of the adaptive function set at that moment. More about this is detailed in Chen and Chie (2003), Section "**Functional-Modularity Approach to Innovation and Knowledge**".

Preference

The second challenging issue is the choice of *the fitness function*, i.e., the *feedback mechanism* by which the direction of technological progress is determined. All commodities associated with their production processes shall be evaluated by *profits*, which are determined in turn by the market demand and the cost structure. The market demand is derived from the users' (consumers') subjective preference (utility function). However, current economic theory provides us no clue on how to evaluate the enjoyment of consuming a commodity when the consumer is presented a sequence of processors. In this paper, a possible solution based on the *monotonicity*, *synergy* and *consistency* condition is proposed to derived a *well-behaved utility function* from any preference which is sampled from the *strongly-typed Kleene Star*. More details will be given in Section "**Functional-Modularity Approach to Preferences and Utility Functions**". This framework can be enriched by making consumers' preference endogenous and adaptive and by taking the effect of *fashion* into account.[3]

Consumers' preferences are also represented by *parse trees* as shown in Figure 5, which can be interpreted as an *ideal* or *targeted* sequence of processors. These six parse trees are different, indicating that consumers' preferences are heterogeneous. Consumers may not explicitly know

[2] As an illustration, a parse-tree representation of the Chinese macaroni may be helpful.

[3] Consumers' preference can be functionally dependent on the commodities they have consumed. Examples are abound. Paul Romer once made the following story, "For instance, cheaper transistors have encouraged broadband graphics applications, which in turn have created users impatient with the slow speed of data transmission." (Romer, 1998).

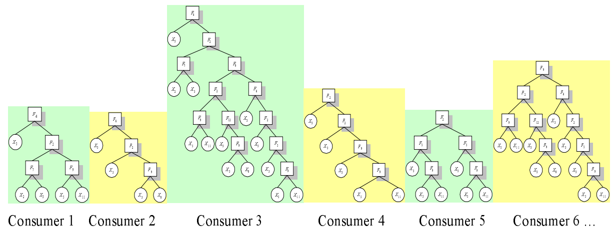

Figure 5: Consumers' Preference: What shown here is only part of the potentially infinite large parse tree, i.e. only U^l of $[U^l]$. See more details on Section "**Functional-Modularity Approach to Preferences and Utility Functions**."

their preferences. However, if the commodity served to them is characterized by exactly the same sequence of processors, consumers shall be very happy about this. The most intriguing issue involved here is the mathematical operation of these preferences. A full discussion of it will come later.

Cost and Capacity Constraint

A full-fledged agent-based model of innovation can be quite complex. As an initial stage of this research, we may start the model-building by making some simplifications. We first assume that a fixed number of consumers (say n_c) whose preferences, represented by a parse tree, are exogenously generated, and whose endowments (income) are also given exogenously. On the production side, the economy is composed of n_f producers, each of them are initially assigned an equally operation capital, K_0.

$$K_{1,0} = K_{2,0} = ... = K_{n_f,0} = K_0. \tag{4}$$

With this initial capital, the producers are able to buy materials and processors from the input markets up to their affordability. There are two types of input markets at the initial stage, namely, the *raw-material market* and the *rudimentary processor market*. For simplicity, we assume that the supply curve of two markets are infinite elastic with a fixed unit cost (c) for each raw materials and for each rudimentary processors.

$$C_{X_1} = C_{X_2} = ... = C_{X_\kappa} = C_{F_1} = C_{F_2} = ... = C_{F_k} = c. \tag{5}$$

With the materials and the rudimentary processors purchased from the input market, the producer can produce a variety of commodities, defined by the associated sequence of processors. The cost of each commodity is then simply its total number of materials and the number of processors, or, in terms of GP, the *node complexity* of the parse tree. However, to allow for the *scale effect*, each additional unit of the same commodity produced by the producer should be less costly. This can be done by introducing a monotonically decreasing function $\tau(q)$ $(0 \leq \tau(q) \leq 1)$, where q is qth unit of the same commodity produced. The cost of each additional unit produced is simply the cost of the previous unit pre-multiplied by $\tau(q)$. With this description, the *capacity constraint* for a *full-specialized producer* i ($i \in [1, ...n_f]$), i.e. the producer who supplies only one commodity, should be

$$K_0 \geq \sum_{q=1}^{\bar{q}} C_q, \tag{6}$$

where $C_q = \tau(q) C_1$ is the unit cost of the qth unit and $\tau(1) = 1$. For a *full-diversified producer*, i.e., the producer who produces a variety of commodities each of which with only one unit, the capacity constraint is

$$K_0 \geq \sum_{m=1}^{\bar{m}} C_{m,1}, \tag{7}$$

where $C_{m,1}$ is the the cost of the first unit of commodity m. In general, the capacity constraint for the producer i is

$$K_0 \geq \sum_{m=1}^{\bar{m}} \sum_{q=1}^{\bar{q}_m} C_{m,q}, \tag{8}$$

where $C_{m,q} = \tau_m(q) C_{m,1}$.

Production Strategies

In Equation (8), the strategic parameters are $\bar{m}$, $\bar{q}$ and C_m. To survive well, producers have to learn how to optimize them. $\bar{m}$ can be be taken as a measure of a degree of *diversification*, whereas $\bar{q}$ can be taken as as a degree of *specialization*. C_m, i.e., the node complexity of the commodity m, is also a behavioral variable. Given the capacity constraint, the producer can choose to supply large amount of primitive commodities (a quantity-oriented strategy), or limited amount of highly delicate commodities (a quality-oriented strategy). Therefore, the choice of C_m can be considered as a choice of the level of *quality*.

Marketing Strategies

The marketing strategy consists of two main stays: *pricing* and *advertising*. On the pricing part, the producer has to decide a *mark-up* η, i.e., the expected profit rate of the commodity. Suppose that $\bar{C}_m$ is the average cost of producing the mth commodity.

$$\bar{C}_m = \frac{\sum_{q=1}^{\bar{q}_m} C_{m,q}}{\bar{q}_m} \tag{9}$$

Then by the associated mark-up η_m, the label price (the *ask*) of the commodity is

$$ask_m = (1 + \eta_m) \bar{C}_m. \tag{10}$$

Advertising can be considered as cost expenditures to subside consumers' search costs (see the discussion of Equation (16) below). It is used to enhance consumers' knowledge of the commodity. Without this expenditure, consumers may not be able to reach this commodity and hence would not buy it. Suppose that the advertising strategy is simply to decide a lump-sum expenditure which is used to cover a portion of consumers' search costs. Let A_m be the advertising expenditure spent for the promotion of commodity, then the capacity constraint for the producer i is

$$K_0 \geq \sum_{m=1}^{\bar{m}} A_m + \sum_{m=1}^{\bar{m}} \sum_{q=1}^{\bar{q}_m} C_{m,q}. \tag{11}$$

Market Process 1: Random Matching

For simplicity, we assume a *random matching mechanism* between consumers and producers. In a trading round t, each consumer i ($i = 1, 2, ..., n_c$) is randomly matched to one producer j ($j = 1, 2, ..., n_f$) and to one of the commodities it produces, say, $Y_{j,m}$. The utility from consuming that commodity, $U_i(Y_{j,m})$, defines the maximum amount (*reservation price*), $B_i(Y_{j,m})$, which the consumer would like to *bid* for that commodity, i.e.,

$$bid_i(Y_{j,m}) = U_i(Y_{j,m}) \tag{12}$$

Let $q^d(Y_{j,m})$ be the *aggregate demand* for the commodity $Y_{j,m}$ at the trading round t,

$$q^d(Y_{j,m}) = \sum_{i=1}^{n_c} I_i, \tag{13}$$

where I_i is an indicator function. $I_i = 1$ if consumer i is connected to the commodity $Y_{j,m}$ at the trading round t and $bid_i(Y_{j,m}) \geq ask(Y_{j,m})$; otherwise, it is zero. Also, let $q^s(Y_{j,m})$ be the total number of units available at the trading round t. The trading price of $Y_{j,m}$ at the trading round t will then be determined as follows.

$$P(Y_{j,m}) = \begin{cases} ask(Y_{j,m}), & if \;\; q^d(Y_{j,m}) \leq q^s(Y_{j,m}) \\ bid_{i^*}(Y_{j,m}), & if \;\; q^d(Y_{j,m}) > q^s(Y_{j,m}) \end{cases} \tag{14}$$

where i^* satisfies the equality

$$r\acute{G}a \quad \{i \mid bid_i(Y_{j,m}) \geq bid_{i^*}(Y_{j,m})\} = q^s(Y_{j,m}). \tag{15}$$

The price determination process (14) and (15) can be considered as a combination of *take-it-or-leave* and *English auction* (ascending-price auction). The sellers just post the price and would basically not change it at the same market day. However, if at a moment the demand is too high, then the seller will leave the consumers to determine where the price shall go. This finishes one trading round. All commodities sold in the trading period t shall be removed from the shops during the next trading period. The matching and trading process goes on and on until either we come to the end of trading day, i.e., at a maximum of T trading rounds or all consumers have run out of their budgets. A flowchart of the market process is given in Figures 6 and 7.

Market Process 2: Purposive Search

A variety of trading processes exists. For example, to decide which commodities to be included into their baskets, consumers can first experiment with different commodities. They can do this by shopping around, and sampling some commodities. However, there is a *search cost* associated with this shopping activity. The unit search cost for experimenting one commodity is a. The total resource spent in search should not be beyond their budget constrains. The consumers can then evaluate the satisfaction they have from each commodity in their sample. With this evaluation, they can determine the reservation price of each commodity. By computing the difference between the reservation price and the label price of the commodity, the net utility of a commodity (consumer surplus) can be derived, and all the commodities in the sample can be ranked accordingly. A rational consumer is expected to buy the commodities starting

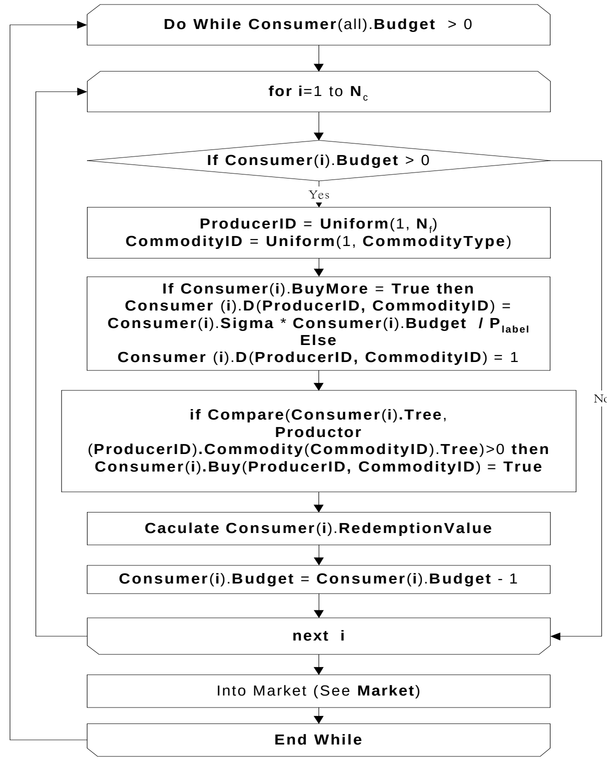

Figure 6: The Flow Chart of Random Matching Mechanism

from the topmost commodity and descending down to the commodities which spent their last penny. Hence, the consumers' budget constraint can be written as

$$I \geq a \times S + \sum_{\tau} P_{\tau}, \tag{16}$$

where I is the budget constraint, S is the sample size (search intensity) of consumers, i.e., the number of commodities consumers have some knowledge of them. P_{τ} is the price of the commodity which is ranked lth of consumers' experienced commodities.

Functional-Modularity Approach to Preferences

Commodity Space

Before introducing the functional-modularity approach to preferences, let us start a brief review on the utility function used in the conventional economic theory. The utility function $U(.)$ is generally a mapping from non-negative real space to real space $\mathcal{R}$.

$$U : \mathcal{R}^n_+ \rightarrow \mathcal{R} \tag{17}$$

This mapping above helps us little when what to evaluate is a sequence of processors rather than just quantity. In our economy, what matters for consumers is not the *quantity* they consumed, but the *quality* they consumed. Therefore, the conventional commodity space $\mathcal{R}^n_+$ is replaced by a new

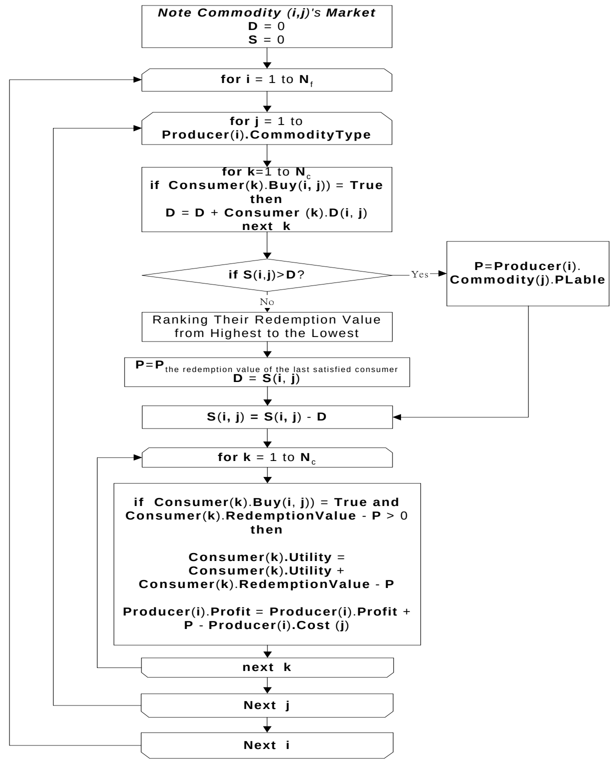

Figure 7: The Flow Chart of the Market Process

commodity space which is a collection of sequences of processors. We shall call the space $\mathcal{Y}$. The representation of the commodity space $\mathcal{Y}$ can be constructed by using *theory of formal language*, for example, the *Backus-Nauer form* (**BNF**) grammar. So, $\mathcal{Y}$ is to be seen simply as the set of all expressions which can be produced from a start symbol Λ under application of *substitution rules* (*grammar*) and a finite set of primitive processors (Σ) and materials (Ξ). That is $\mathcal{Y}$ represents the set of all commodities which can be produced from the symbols of Σ and Ξ.

$$\mathcal{Y} = \{Y \mid \Lambda \Rightarrow Y\} \tag{18}$$

While, as we saw in Figure 5, each Y ($Y \in \mathcal{Y}$) can be represented by the language of expression trees (**ETs**), a more effective representation can be established by using *Gene Expression Programming* (**GEP**), developed by Ferreira (2001). In GEP the individuals are encoded as *linear strings of fixed length* (the genome or chromosomes) which are afterwards expressed as nonlinear entities of different sizes and shapes, i.e., different expression trees. As Ferreira (2001) shown, the interplay of chromosomes and expression trees in GEP implies an unequivocal translation system for translating the language of chromosomes into the language of ETs. Some advantages of GEP over genetic programming and genetic algorithms has been well discussed in Ferreira (2001). Using GEP, the commodity space can then be defined as a subset of *Kleene star*, namely,

$$\mathcal{Y} = \{Y_n \mid Y_n \in (\Sigma \cup \Xi)^* \cap GEP\}, \tag{19}$$

where Y^n is a string with length n,

$$Y_n = y_1 y_2 ... y_n, \;\; y_i \in (\Sigma \cup \Xi), \forall i = 1, ..., n. \tag{20}$$

We have to emphasize that, for satisfying the syntactic validity, $\mathcal{Y}$ is only a subset of the Kleene star $(\Sigma \cup \Xi)^*$. To make this distinction, the $\mathcal{Y}$ described in (19) is referred to as the *strongly-typed Kleene star*. Each Y_n can then be translated into the familiar parse tree by using GEP. This finishes our description of the commodity space.

Preferences

Unlike commodity space, preference space cannot be a collection of finite-length strings, since they are not satisfied with the *non-saturation* assumption. Economic theory assumes that consumer always prefer more to less, i.e., the marginal utility can never be negative.

$$U'(y) \geq 0, \forall y \in \mathcal{R}_+ \tag{21}$$

Even though we emphasize *quality* dimension instead of *quantity* dimension, a similar vein of (21) should equally hold: *you will never do enough to satisfy any consumer*. If consumers' preferences are represented by finite-length strings, then at a point, they may come to a state of complete happiness, known as the *bliss point* in economic theory. From there no matter how hard the producers try to upgrade their existing commodities, it is always impossible to make consumers feel happier. This is certainly not consistent with our observation of human behavior. As a result, the idea of commodity space cannot be directly extended to preference space.

To satisfy the non-saturation assumption, preference must be a string with infinite length, something like

$$...u_1 u_2 ... u_l ... = ...U^l ... \tag{22}$$

However, by introducing the symbol ∞, one can regain the finite-length representation of the preference, i.e.,

$$\infty u_1 u_2 ... u_l \infty = \infty U^l \infty = [U^l]. \tag{23}$$

First of all, as we mentioned earlier, consumers may not necessarily know what their preferences look like, and may not even care to know it. However, from Samuelson's *revealed preference theory*, we know that consumers' preferences *implicitly* exist. Equation (23) is just another way to say that consumers' preferences are *implicit*. It would be pointless to write down the consumers' preferences of the 30 century, while we may know that there are much richer than what has been revealed today. To approximate the feedback relation between technology advancements and preferences, it would be good enough to work with *local-in-time* preferences (temporal preference).

Secondly, Equation (23) makes us be able to see the possibility that preference is adaptive, evolving and growing. What will appear in those ∞ portions may crucially depends on the commodities available today, commodities consumed by the consumer, consumption habits of other consumers, and other social, institutional and scientific considerations.

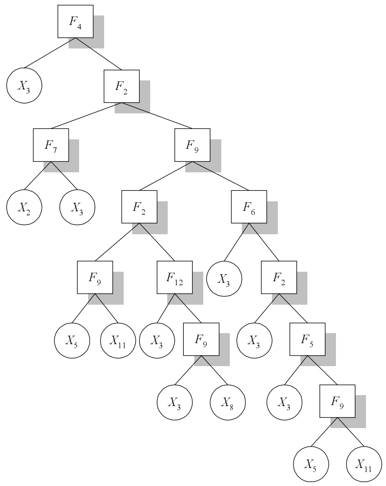

Figure 8: Preference: The Parse-Tree Representation

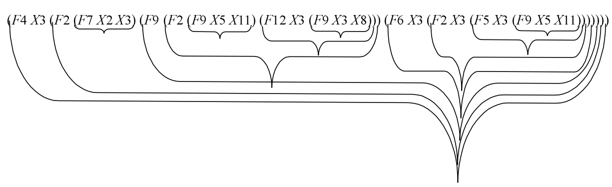

Figure 9: Modular Preference: The LISP Representation

Utility Function

Given the preference $[U^l]$, let $U \mid [U^l]$ be the utility function derived from $[U^l]$. $U \mid [U^l]$ is a mapping from the *strongly-typed Kleene Star* to $\mathcal{R}_+$.

$$U \mid [U^l] : \mathcal{Y} \mathcal{R} \not\to {}_+ . \qquad (24)$$

Hereafter, we shall simply use U instead of $U \mid [U^l]$ so long as it causes no confusion.

The modular approach to preference considers each preference as a hierarchy of modular preferences. Each of these modular preferences is characterized by a parse tree or the so-called building block. For example, the preference shown in Figure 8 can be decomposed into modular preferences with different depths. They are all explicitly indicated in Figure 9. Consider S_i as the set of all modular preferences with *depth* i. Then Table 1 lists all modular preferences by these S_i. From both Figure 9 and Table 1, it is clear that each subtree at a lower level, say S_j, can always finds its parent tree, a thee which it is a part or a branch of it, at a higher level, say S_i where $i > j$. This subsequence relation can be represented as follows.

$$S_i \sqsupseteq S_j \qquad (25)$$

A commodity Y_n is said to *match* a modular preference S_i of U^l if they are exactly the same, i.e. they share the same the LISP expression and the same tree representation. Now, we are ready to postulate the first regularity condition about a well-behave utility function, which is referred to as

Table 1: Modular Preferences Sorted by Depth

D	Subtrees or terminals	
1	$X_2, X_3, X_5, X_8, X_9, X_{11}$	1
2	$S_{2,1} = (F_7 X_2 X_3)$ $S_{2,2} = (F_9 X_5 X_{11})$ $S_{2,3} = (F_9 X_3 X_8)$ $S_{2,4} = (F_9 X_5 X_{11})$	2
3	$S_{3,1} = (F_{12} X_3 (F_9 X_3 X_8))$ $S_{3,2} = (F_5 X_3 (F_9 X_5 X_{11}))$	4
4	$S_{4,1} = (F_2 (F_9 X_5 X_{11})(F_{12} X_3 (F_9 X_3 X_8)))$ $S_{4,2} = (F_2 X_3 (F_5 X_3 (F_9 X_5 X_{11})))$	8
5	$S_5 = (F_6 X_3 (F_2 X_3 (F_5 X_3 (F_9 X_5 X_{11}))))$	16
6	$S_6 = (F_9 (F_2 (F_9 X_5 X_{11})(F_{12} X_3 (F_9 X_3$ $X_8)))(F_6 X_3 (F_2 X_3 (F_5 X_3 (F_9 X_5 X_{11})))))$	32
7	$S_7 = (F_2 (F_7 X_2 X_3)(F_9 (F_2 (F_9 X_5 X_{11})(F_{12}$ $X_3 (F_9 X_3 X_8)))(F_6 X_3 (F_2 X_3 (F_5 X_3$ $(F_9 X_5 X_{11}))))))$	64
8	$S_8 = (F_4 X_3 (F_2 (F_7 X_2 X_3)(F_9 F_2 (F_9 X_5 X_{11})$ $(F_{12} X_3 (F_9 X_3 X_8)))(F_6 X_3 (F_2 X_3 (F_5 X_3$ $(F9 X_5 X_{11})))))))$	128

the *monotonicity condition*.

> *Monotonicity*
> Given a preference $[U^l]$, the associated utility function is said to satisfy the *monotonicity condition* iff
>
> $$U(Y_{n_i}) > U(Y_{n_j}) \qquad (26)$$
>
> where Y_{n_i} and Y_{n_j} are the commodity matching the corresponding modular preferences S_i and S_j of U^l and S_i and S_j satisfy Equation (25).

The *monotonicity* condition can be restated in a more general way.

> *Monotonicity*
> Given a preference $[U^l]$ and let $\{h_1, h_2, ...h_j\}$ be an increasing subsequence of $\mathcal{N}_+$, then the associated utility function is said to satisfy the *monotonicity condition* iff
>
> $$U(Y_{n_j}) > U(Y_{n_{j-1}}) >..> U \quad (Y_{n_2}) > U(Y_{n_1}) \qquad (27)$$
>
> where $Y_{n_1}, ..., Y_{n_j}$ are the commodity matching the corresponding modular preferences $S_{h_1}, ..., S_{h_j}$ of U^l, and
>
> $$S_{h_i} \sqsupseteq S_{h_{i-1}} \sqsupseteq ... \sqsupseteq S_{h_2} \sqsupseteq S_{h_1}. \qquad (28)$$

If S_j is a subtree of S_i as in Equation (25), then S_k is called the *largest subtree* of S_i if S_k is a *branch* (descendant) of S_i. We shall use "$S_i \triangleleft S_k$" to indicate this largestmember relation. Depending on the grammar which we use, the largest subtree of S_i may not be unique. For example, each modular preference in Figure 8 has two largest subtrees. In general, let $S_{h_1}, S_{h_2}, ...S_{h_n}$ be all the largest subtrees of Y_i, denoted as follows:

$$S_i = \sqcup_{h_1}^{h_j} S_k \lhd \{S_{h_1}, S_{h_2}, ...S_{h_j}\}, \tag{29}$$

where $\{h_1, h_2, ...h_j\}$ is a non-decreasing subsequence of $\mathcal{N}_+$. Notice these largest trees must not have sub-relation (25) among each other. However, they may have different depths, and the sequence $\{h_1, h_2, ...h_j\}$ ranks them by depth in an ascending order so that S_{h_1} is the largest subtree with the minimum depth, and S_{h_j} is the one with the maximum depth.

The second postulate of the well-behave utility function is the property known as *synergy*.

> *Synergy:*
> Given a preference $[U^l]$, the associated utility function is said to satisfy the *synergy condition* iff
>
> $$U(Y_{n_i}) \geq \sum_{k=1}^{j} U(Y_{n_k}), \tag{30}$$
>
> where Y_{n_i} and $\{Y_{n_k}; k = 1, ..., j\}$ are the commodity matching the corresponding modular preferences S_i and $\{S_{h_k}; k = 1, ..., j\}$ of $[U^l]$ and S_i, and $\{S_{h_k}; k = 1, ..., j\}$ satisfies Equation (29).

For convenience, we shall also the notation $\sqcup_{k=1}^{j} Y_{n_k}$ as the synergy of the set of commodities $\{Y_{n_k}; k = 1, ..., j\}$. Based on the *New Oxford Dictionary of English*, synergy is defined as "'the interaction or cooperation of two or more organizations, substances, or other agents to produce a combined effect greater than the sum of their separate effects". "*The whole is greater than the sum of the parts*" is the fundamental source for *business value creation*. Successful business value creation depends on two things: *modules* and the *platform* to combine these modules. Consider the consumer characterized by Figure 8 as an example. To satisfy him, what needed are all of the modules listed in Table 1. Even though the technology has already advanced to the level S_7, knowing the use of processor F_4 to combine X_3 and S_7 can still satisfy the consumer to a higher degree, and hence creating a greater business value. More of this will be discussed on Chen and Chie (2003), Section " **Knowledge Market**".

A modular preference may appear many times in a preference. For example, $S_{2,4}$ in Table 1 appears twice in Figure 8. In this case, it can simultaneously be the largest subtree of more than one modular preference. For example, $S_{2,4}$ is the largest subtree of both $S_{3,2}$ and $S_{4,1}$. Let S_k be the largest subtree of S_{h_1}, S_{h_2},..., and S_{h_j}. Denote this relation as

$$S_k = \sqcap_1^j S_{h_i} \rhd \{S_{h_1}, S_{h_2}, ...S_{h_j}\}. \tag{31}$$

> *Consistency:*
> Given a preference $[U^l]$, the associated utility function is said to satisfy the *consistent condition* iff
>
> $$U(Y_{n_i} \mid S_k \rhd S_{h_1}) = ... = U(Y_{n_i} \mid S_k \rhd S_{h_j}), \tag{32}$$
>
> where $Y_{n_i} \mid S_k \rhd S_{h_1}$ is the commodity which matches the corresponding modular preference S_k in the designated position, $S_k \rhd S_{h_i}$.

The consistency condition reiterates the synergy effect. No matter how intensively the commodity Y_{n_i} may contribute significantly to the value creation of a synergy commodity, its value will remain identical and lower when it is served alone.

> *Well-Behaved:*
> Given a preference $[U^l]$, the associated utility function U is said to be *well-behaved* iff it satisfy the monotone, synergy and consistency condition. It generates a sequence of number $\{U(Y_{n_i})\}_{i=1}^{h}$ where Y_{n_i} matches the respective modular preference $S_{d,j}$. $S_{d,j}$ is the jth modular preference with depth d.

The utility assigned in Table 1 is an illustration of a well-behaved utility function derived from the preference shown in Figure 8. In fact, this specific utility function is generated by the following exponential function with base 2.

$$U(S_{d,j}) = 2^{d-1} \tag{33}$$

Utility function (33) sheds great light on the synergy effect. So, primitive materials or rudimentary commodities may only satisfy the consumer to a rather limited extent. However, once after suitable processing or integration, their value can become increasingly large to the consumer. The exponential function with base 2 simply shows how fast the utility may be scaled up, and hence may provide a great potential incentive for producers to innovate. Of course, to be a well-behaved utility function, U can have many different function forms. Some of them may have good economic intuitions, and some may not.

Module Matching

Now, it is high time to answer the question: *what would be the enjoyment for a consumer with a preference $[U^l]$ consuming a commodity Y_i?* Let us start tackling this issue by a commodity, called the *simple commodity*. Given a preference $[U^l]$, a commodity Y_i is called *simple* with respect to $[U^l]$ if it matches exactly one modular preference of U^l. It is easy to evaluate the simple commodity, as discussed in the previous section and exemplified in Table 1.

However, not all commodities are simple. Y_i, as a whole, may match none of any modular preference of $[U^l]$. Nevertheless, it can be still enjoyable for the consumer if it is *similar* or *close* to consumer preference $[U^l]$ in many regards. In this section, we propose an evaluation scheme based on a

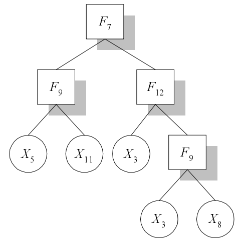

Figure 10: An Example of Commodity: The LISP Representation

idea of "*similarity*" or "closeness". The evaluation scheme is called *module matching*.

The idea of modular matching is very straightforward. As what now should becomes clear, each *commodity* is composed of many *modular commodities* with different depths. For example, the commodity represented in Figure 10 has a list of modular commodities as shown in Table 2. Let $Y_{d,j}$ be the jth modular commodity with depth d. Now let the commodity be presented to the consumer with a preference as depicted in Figure 8. Clearly, $Y_{4,1}$ does not match any modular preference as listed in Table 1. However, the commodity is similar to consumer's preference since it has a major part, $Y_{3,1}$, which matches consumer preference exactly by the modular $S_{3,1}$ (Table 1). Therefore, while the commodity is not the same as consumer's preference, it is not irrelevant and can satisfy the consumer to some degree.

Table 2: Modular Commodities

d	Subtrees or terminals
1	X_3, X_5, X_8, X_{11}
2	$Y_{2,1} = (F_9 X_5 X_{11})$ $Y_{2,2} = (F_9 X_3 X_8)$
3	$Y_{3,1} = (F_{12} X_3 (F_9 X_3 X_8))$
4	$Y_{4,1} = (F_7 (F_9 X_5 X_{11})(F_{12} X_3 (F_9 X_3 X_8)))$

Next, let us take away $Y_{3,1}$ from the commodity Y_i, what left is only the subtree corresponding to $Y_{2,1}$ (Figure 10). This part also matches the the consumer preference by the modular $S_{2,2}$. The value of the commodity to the consumer would, therefore, be enhanced as opposed to the case if $Y_{2,1}$ were completely useless. We then take $Y_{2,1}$ away from Y_i, and there is nothing left. So, the set of modular preference matched by Y_i is

$$M_{Y_i:[U^l]} = \{S_{3,1}, S_{2,2}\}. \tag{34}$$

Therefore, the utility of the commodity Y_i with respect to the preference $[U^l]$ can be written as

$$U(Y_i) = U(S_{3,1}) + U(S_{2,2}) = 2 + 4 = 6 \tag{35}$$

It is crucial to make some working principles underlying this example explicitly. First, we do not start module-matching from the smallest modules, such as X_3, X_5, X_8 and X_{11} (Table 1). Instead, we start from the biggest one, i.e, the one with the maximum depth. This doing is referred as to the *descending principle*. Secondly, once any modular commodity is shown to match the corresponding modular preference, it is no longer usable for the rest of matching exercise. This is called the *non-redundancy principle*. The working of these tow principles excludes the consideration of the modular commodity $Y_{2,2}$, while it also matches the preference by $S_{2,3}$.

The main purpose of these two principles is to avoid *double-counting* and simultaneously to derive the maximum value of the respective commodity. For example, if one start the modular-matching in an *ascending* order, then after the matches of the four raw materials X_3, X_5, X_8 and X_{11}, nothing left. Hence the utility of Y_i would come up with only 4, which obviously fails to take the synergy effect into account.

The module-matching algorithm is summarized as follows.

- Step 1: List all modular commodities of Y_i in a collection $\mathcal{C}_{Y_i}$, and group them by depth, say from $d = 1, 2, ..d_{max}$.
- Step 2: Set $d = d_{max}$. Start the modular-matching from $Y_{d_{max}} (= Y_i)$. If there is a match, which means Y_i is a *simple commodity*, then set

  $$U(Y_i) = U(S_{i,j}), \tag{36}$$

  where $S_{i,j}$ is the modular preference matched by $Y_{d_{max}} = Y_i$. Go to step 7. If there is no match, go to the next step.
- Step 3: Decrease d by 1, and do modular-matching for $Y_{d,j}, \forall j$.
- Step 4: For any match, Y_{d,j^*}. Delete all its modular commodities from $\mathcal{C}_{Y_i}$.
- Step 5: Put all matches into the set $M_{Y_i:[U_l]}$.
- step 6: If $d = 0$, or $\mathcal{C}_{Y_i}$ is a null set, then

  $$U(Y_i) = \sum_{S_{d,j} \in M_{Y_i:[U_l]}} U(S_{d,j}); \tag{37}$$

  otherwise, go back to step 3.
- step 7: Stop.

Concluding Remarks

This paper proposes a functional-modularity approach to economic models of innovation within an agent-based computational modeling context. It is motivated by a series of former applications of genetic programming to knowledge discovery and data mining in the area of finance, sciences and engineering. In these applications, two crucial features of innovation was demonstrated via genetic programming, namely, *evolving* and *growing*. In some engineering applications, the evolving and growing process was actually displayed via the change of the outer topology or the inner structure of real entities. This progress makes it possible to build a *direct* modeling, observation and measure of innovation processes.

However, studying economic activities of innovation has a much broader scope than just an innovation itself. It is concerned with the incentive to innovate, the resources used to support an innovation, the success, the lifespan, and the distribution of an innovation. It is also inextricably interwoven with the evolution of human preferences and culture. The associated social impacts, such as the wealth distribution, the growth of knowledge capital, and market structure are also important considerations. To be able to have this broader view, an agent-based computational economic model of innovation is proposed in this study.

In this agent-based model of innovation, breakthroughs are made in several fundamental elements of economics, which include a functional-modularity re-formulation of commodities, production, preference and technology (knowledge). Modular preferences, modular commodities, and modular technologies become the main working concepts of this economy. Breakthroughs are also made via the use of *Gene Expression Programming* to characterize the commodity space as a *strongly-typed* Kleene star. Axioms of monotonicity, synergy, and consistency are introduced to define a well-behaved utility function associated with a given preference. The distinguishing feature of a knowledge-based economy is particularly highlighted by the synergy axiom or the synergy effect. The utility of consuming a specific commodity is solved by using an algorithm based on module matching.

With this fundamental re-formulation, market mechanism and producers' adaptation are operated accordingly. Two markets are considered in this model, namely the commodity market and the knowledge market. In the commodity market, a number of producers are competing for a number of consumers whose preferences are randomly generated initially but may change over time. To shape their competitiveness, producers have to make their critical strategies ranging from production, marketing to R&D. The last one decides their involvement to the knowledge market where one can open and acquire promising modular technologies.

Genetic programming is applied to simulate the evolution of technologies within this agent-based context. In addition to technology, producers' competition strategies will also evolve with time and that evolution is mainly driven by the survival pressure.

References

Allen, F. and R. Karjalainen, (1999). "Using Genetic Algorithms to Find Technical Trading Rules." *Journal of Financial Economics,* **51(2)** 245–271.

Aoki, M., (2002a). "Open Models of Share Markets with Two Dominant Types of Participants," *Journal of Economic Behavior and Organization*, **49** 199–216.

Aoki, M., (2002b). "Applications of Ewens–Pitman–Zabell Inductive Methods in New Economic Dynamics," *Proceeding of Sixth International Conference on Complex Systems*, 29–35.

Banville, J., (1993). *Kelper: A Novel (Vintage International)*. Vintage Books.

Basalla, G., (1988). *The Evolution of Technology*. Cambridge University Press.

Chen, S.-H. and C.-H. Yeh (2001). "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**, 363–393.

Chen, S.-H. and B.-T. Chie (2003). "Toward A Functional-Modularity Approach to Knowledge-Based Economy," *The 3rd European Meeting of Applied Evolutionary Economics* (**EMAEE'03**).

Ferreira, C., (2001). "Gene Expression Programming: A New Adaptive Algorithm for Solving Problems," *Complex Systems*, **13 (2)**, 87-129.

Keber W. and N. J. Saam, (2001). "Competition as a Test of Hypotheses: Simulation of Knowledge-generating Market Processes." *Journal of Artificial Societies and Social Simulation*, **4** (3), http://www.soc.surrey.ac.uk/JASSS/4/3/2.html.

Levy, S., (1992). *Artificial Life: A Report from the Frontier Where Computers Meet Biology*, Vintage, NewYork.

Ma, T. and Y. Nakamori, (2002). "An Agent-Based Simulation on Dynamics of Design Spaces," *Proceeding of The Second International Workshop on Agent-based Approaches in Economic and Social Complex Systems (AESCS'02)*, 135–142.

Neely, C. and P. Weller, (1999). "Technical Trading Rules in the European Monetary System." *Journal of International Money and Finance*, **18** 429–458.

Neely, C., P. Weller, and R. Dittmar, (1997). "Is Technical Analysis in the Foreign Exchange Market Profitable? a Genetic Programming Approach." *Journal of Financial and Quantitative Analysis*, **32(4)** 405–426.

Willihnganz, A., (1999). "Software that Writes Software," *salon.com*, WWW Article.

Zabell, S., (1992). "Predicting the Unpredictable," *Synthese*, **90** 205–232.