International Journal of Modern Physics B
Vol. 18, Nos. 17–19 (2004) 2376–2386

FUNCTIONAL MODULARITY IN THE FUNDAMENTALS OF ECONOMIC THEORY: TOWARD AN AGENT-BASED ECONOMIC MODELING OF THE EVOLUTION OF TECHNOLOGY*

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

BIN-TZONG CHIE
AI-ECON Research Center, Department of Economics,
National Chengchi University, Taipei, 116, Taiwan
chie@aiecon.org

Received 31 August 2003

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.

Keywords: Agent-based computational economics; innovation; functional modularity; genetic programming.

1. Motivation and Introduction

No matter how commonly the term "*innovation*" or "*technological progress*" has been used in economics, or more generally, in the 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 lacking; however, the *general representation* of technology, based on which innovation can be defined and its evolutionary process studied, does not exist.

*An early version of the paper was presented at the *9th International Conference on Computing in Economics and Finance* and the *First Cross-Strait Conference on Statistical Physics*. It was also presented as a part of a sequence of intensive lectures given at the University of Science and Technology of China (USTC). The authors are grateful to Prof. Bing-Hong Wang for hosting the authors' visit at USTC. The authors would also like to thank the two anonymous referees for their helpful suggestions. NSC research grant No. 92-2415-H-004-005 is gratefully acknowledged.

While direct modeling of innovation is difficult, economists' dissatisfaction with the neo-classical economic research paradigm is increasing, partially due to its incompetence in terms of producing novelties (or the so-called *emergent property*). We cannot assume in advance that we know all new goods and new technology that will to be invented in the future. Therefore, in our model, we must leave space to anticipate the unexpected. Recently, Aoki[2,3] introduced Zabell's notion of *unanticipated knowledge* to economists.[10] This notion is motivated by *population genetics.* In probability and statistics it is referred to as the *law of succession*, i.e. how to specify the conditional probability that the next sample is never seen, given available sets of observations up to now. However, the Ewens–Pitman–Zabell induction method proposed by Aoki is still rather limited. Basically, the nature of diversity of species and the nature of human creativity should not be treated equally.[5]

This paper proposes *genetic programming* as a possible approach leading to simulating the evolution of technology. Our argument is based on two essential standpoints. First of all, as regards the innovation process, we consider it to be a *continuous process* (*evolution*), rather than a *discontinuous process* (*revolution*). According to the continuity hypothesis, novel artifacts can only arise from antecedent artifacts. Second, the evolution can be regarded as a *growing process* by 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. This feature, known as *functional modularity*, can be demonstrated by GP, and that will be shown in this paper.

2. Background

The idea of functional modularity is not new to economists. For example, Paul Romer has already mentioned that "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). That GP can deliver this feature has already been well evidenced in a series of promising applications to the scientific, engineering, and financial domain.

A decade ago, financial economists started to apply the functional-modularity approach with GP to discover *new* trading rules. Neely, Weller and Dittmar[8] and Allen and Karjalainen[1] 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. Hence, GP already demonstrated the evolution of trading technology: 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 corresponded to an earlier conjecture by Kepler, published ten years before the great mathematician finally perfected the equation.[4,7] A further application of GP by John Koza to analog circuits shows that GP-evolved solutions can actually compete with human ingenuity: the results have closely matched ideas contrived by humans. Koza's GP has produced circuit designs that infringe 21 patents in all, and duplicate the functionality of several others in novel ways.[9]

3. Commodities and Production

Commodities in economic theory are essentially empty in terms of content. Little attention has been paid to their size, shape, topology, and inner structure. A general representation of commodities simply does not exist in current economic theory. In this paper, each commodity is associated with its *production process*. Each production process is described by a sequence of processors and the materials employed. In general, each sequence may be further 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, we can ascertain the two major elements of GP, namely, the *function set* and the *terminal set*. The former naturally refers to a set of *primitive processors*, whereas the latter refers to a set of *raw materials*. They are denoted respectively by the following,

$$\text{Function Set: } \Xi = \{F_1, F_2, \ldots, F_k\}, \tag{1}$$

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

Each sequence (commodity, processor) can then be represented by a *LISP S-expression* or, simply, *a parse tree* (Fig. 1). The evolution of production processes (commodities) can then be simulated by using standard GP. The *knowledge capital* of the society at a point in time can then be measured by the complexity and the diversity of its existing production processes.

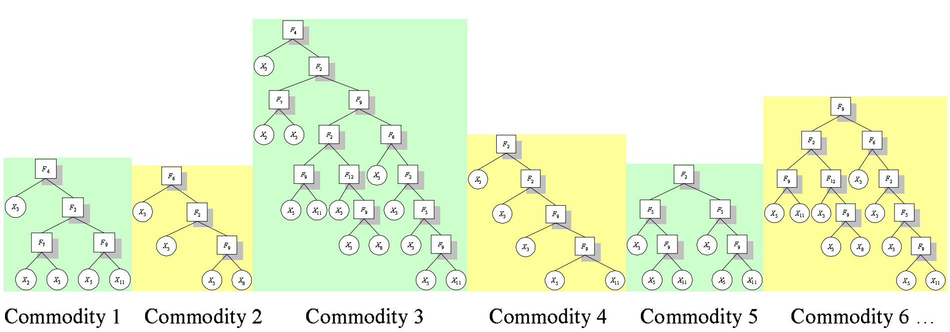

Fig. 1. A functional-modularity representation of commodities. Commodities are associated with their respective production processes which, when written in LIST programming language, can be depicted as parse trees as shown here.

4. Commodity Space

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

$$U : \mathcal{R}_{+}^{n} \to \mathcal{R}\,. \tag{3}$$

This above mapping is of little help to us when what we evaluate is a sequence of processors rather than just a quantity. In our economy, what matters to consumers is not the *quantity* they consumed, but the *quality* of what they consumed. Therefore, the conventional commodity space $\mathcal{R}_{+}^{n}$ is replaced by a new 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 the *theory of formal language*, for example, the *Backus–Nauer form* (BNF) of 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 an 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 Σ and Ξ.

$$\mathcal{Y} = \{Y \mid \Lambda \Rightarrow Y\}\,. \tag{4}$$

While, as we saw in Fig. 1, 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.[6] In GEP the individuals are encoded as *linear strings of fixed length* (the genome or set of chromosomes) which are afterwards expressed as nonlinear entities of different sizes and shapes, i.e. different expression trees. As Ferreira[6] showed, 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. By using GEP, the commodity space can then be defined as a subset of the *Kleene star*, namely,

$$\mathcal{Y} = \{Y_n \mid Y_n \in (\Sigma \cup \Xi)^* \cap \text{ GEP}\}\,, \tag{5}$$

where Y_n is a string of length n,

$$Y_n = y_1 y_2 \cdots y_n\,, \quad y_i \in (\Sigma \cup \Xi)\,, \quad \forall\, i = 1, \dots, n\,. \tag{6}$$

We have to emphasize that, in order to satisfy 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 Eq. (5) 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 ends our description of the commodity space.

5. Preferences

Unlike a commodity space, a preference space cannot be a collection of finite-length strings, since they are not satisfied by the *non-saturation* assumption. Economic theory assumes that consumers always prefer more to less, i.e. the marginal utility can never be negative. Even though we emphasize the *quality* dimension instead of the *quantity* dimension, a similar vein 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 a commodity space cannot be directly extended to a preference space.

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

$$\cdots u_1 u_2 \cdots u_l \cdots = \cdots U^l \cdots . \tag{7}$$

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

$$\infty u_1 u_2 \cdots u_l \infty = \infty U^l \infty = [U^l] . \tag{8}$$

First of all, as we mentioned earlier, consumers may not necessarily know what their preferences look like, and may not even care to know. However, from Samuelson's *revealed preference theory*, we know that consumers' preferences *implicitly* exist. Equation (8) is just another way of saying that consumers' preferences are *implicit.* It would be pointless to write down the consumers' preferences of the 30th century, even though we may know that these 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 preferences).

Secondly, Eq. (8) makes us able to see the possibility that preference is adaptive, evolving and growing. What will appear in those ∞ portions may crucially depend on the commodities available today, the commodities consumed by the consumer before, the consumption habits of other consumers, and other social, institutional and scientific considerations.

6. 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 $\mathcal{R}_+$.

$$U \mid [U^l] : \mathcal{Y} \rightarrow \mathcal{R}_+ . \tag{9}$$

Hereafter, we shall simply use U instead of $U \mid [U^l]$ as 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 Fig. 2 can

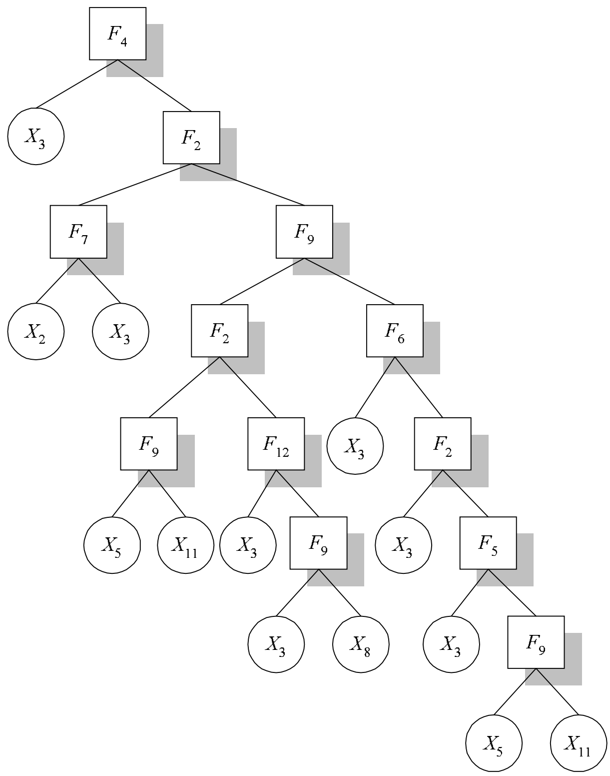

Fig. 2. Preference: The Parse-tree representation. What is shown here is only part of the potentially infinitely large parse tree, i.e. only U^l of $[U^l]$.

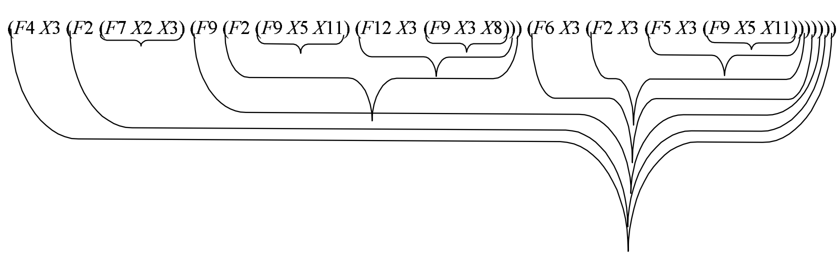

Fig. 3. Modular Preference: The LISP representation.

be decomposed into modular preferences of different depths. They are all explicitly indicated in Fig. 3. Consider S_i to be the set of all modular preferences of *depth* i. Then Table 1 lists all modular preferences by means of these S_i. From both Fig. 3 and Table 1, it is clear that each subtree at a lower level, say S_j, can always find its parent tree, of which it is a part, at a higher level, say S_i where $i > j$. This subsequence relation can be represented as follows:

$$S_i \sqsupset S_j. \tag{10}$$

Table 1. Modular preferences sorted by depth.

$D(d)$	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	$S_6 = (F_9 (F_2 (F_9 X_5 X_{11})(F_{12} X_3 (F_9 X_3 X_8)))$ $\times (F_6 X_3 (F_2 X_3 (F_5 X_3 (F_9 X_5 X_{11})))))$	32
7	$S_7 = (F_2 (F_7 X_2 X_3)(F_9 (F_2 (F_9 X_5 X_{11})$ $\times (F_{12} X_3 (F_9 X_3 X_8)))(F_6 X_3 (F_2 X_3 (F_5 X_3 (F_9 X_5 X_{11}))))))$	64
8	$S_8 = (F_4 X_3 (F_2 (F_7 X_2 X_3)(F_9 F_2 (F_9 X_5 X_{11})$ $\times (F_{12} X_3 (F_9 X_3 X_8)))(F_6 X_3 (F_2 X_3 (F_5 X_3 (F9 X_5 X_{11})))))))$	128

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 regarding a well-behaved utility function, which is referred to as the *monotonicity condition.*

Given a preference $[U^l]$, the associated utility function is said to satisfy the *monotonicity condition* iff

$$U(Y_{n_i}) > U(Y_{n_j}) \tag{11}$$

where Y_{n_i} and Y_{n_j} are the commodities matching the corresponding modular preferences S_i and S_j of U^l and S_i and S_j satisfy Eq. (10).

The *monotonicity* condition can be restated in a more general way. Given a preference $[U^l]$ and by letting $\{h_1, h_2, \ldots, h_j\}$ be an increasing subsequence of $\mathcal{N}_+$, then the associated utility function is said to satisfy the *monotonicity condition* iff

$$U(Y_{n_j}) > U(Y_{n_{j-1}}) > \cdots > U(Y_{n_2}) > U(Y_{n_1}) \tag{12}$$

where $Y_{n_1}, \ldots, Y_{n_j}$ are the commodities matching the corresponding modular preferences $S_{h_1}, \ldots, S_{h_j}$ of U^l, and

$$S_{h_i} \sqsupset S_{h_{i-1}} \sqsupset \cdots \sqsupset S_{h_2} \sqsupset S_{h_1}\,. \tag{13}$$

If S_k is a subtree of S_i as in Eq. (10), then S_k is called the *largest subtree* of S_i if S_k is a *branch* (descendant) of S_i. We shall use "$S_i \lhd S_k$" to indicate this largest-member relation. Depending on the grammar which we use, the largest subtree of S_i may not be unique. For example, each modular preference in Fig. 2 has two largest subtrees. In general, let $S_{h_1}, S_{h_2}, \ldots, S_{h_j}$ be all the largest subtrees of S_i, denoted as follows:

$$S_i = \sqcup_{h_1}^{h_j} S_k \lhd \{S_{h_1}, S_{h_2}, \ldots, S_{h_j}\}, \tag{14}$$

where $\{h_1, h_2, \ldots, h_j\}$ is a non-decreasing subsequence of $\mathcal{N}_+$. Notice these largest trees may not have sub-relationships (10) among each other. However, they may have different depths, and the sequence $\{h_1, h_2, \ldots, h_j\}$ ranks them by depth in an ascending order so that S_{h_1} is the largest subtree with minimum depth, and S_{h_j} is the one with maximum depth.

The second postulate of the well-behaved utility function is the property known as *synergy*. Given a preference $[U^l]$, the associated utility function is said to satisfy the *synergy condition* iff

$$U(Y_{n_i}) \geq \sum_{k=1}^{j} U(Y_{n_k}), \tag{15}$$

where Y_{n_i} and $\{Y_{n_k}; k = 1, \ldots, j\}$ are the commodities matching the corresponding modular preferences S_i and $\{S_{h_k}; k = 1, \ldots, j\}$ of $[U^l]$, and S_i and $\{S_{h_k}; k = 1, \ldots, j\}$ satisfy Eq. (14).

For convenience, we shall also use the notation $\sqcup_{k=1}^{j} Y_{n_k}$ as the synergy of the set of commodities $\{Y_{n_k}; k = 1, \ldots, j\}$. 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 Fig. 2 as an example. To satisfy him, what is 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 create a greater business value.

A modular preference may appear many times in a preference. For example, $S_{2,4}$ in Table 1 appears twice in Fig. 2. 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 $S_{3,2}$ aned $S_{4,1}$. Let S_k be the largest subtree of $S_{h_1}, S_{h_2}, \ldots,$ and S_{h_j}. Denote this relation as

$$S_k = \sqcap_1^j S_{h_i} \rhd \{S_{h_1}, S_{h_2}, \ldots, S_{h_j}\}. \tag{16}$$

Given a preference $[U^l]$, the associated utility function is said to satisfy the *consistent condition* iff

$$U(Y_{n_i} \mid S_k \rhd S_{h_1}) = \cdots = U(Y_{n_i} \mid S_k \rhd S_{h_j}), \tag{17}$$

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 significantly contribute to the value creation of a synergy commodity, its value will remain identical and lower when it is served *alone*.

Given a preference $[U^l]$, the associated utility function U is said to be *well-behaved* iff *it satisfies the monotone, synergy and consistency condition.* It generates a sequence of numbers $\{U(Y_{n_i})\}_{i=1}^{h}$ 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 Fig. 2. In fact, this specific utility function is generated by the following exponential function with base 2.

$$U(S_{d,j}) = 2^{d-1}\,. \tag{18}$$

Utility function (18) sheds great light on the synergy effect. Thus, primitive materials or rudimentary commodities may only satisfy the consumer to a rather limited extent. However, once suitable processing or integration takes place, 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 functional forms.

7. Concluding Remarks and Future Work

In this paper, commodities, production and preference, those fundamentals of economic theory, have been re-formulated in light of functional modularity. We believe that this re-formulation work is original and productive. It lays the foundation upon which one can build and simulate the evolution of technology, more specifically, within the context of agent-based computational economic (ACE) models. A full picture of this ACE model has not been presented in this paper, partially due to the limitations of size imposed on the paper. We, therefore, can only give a sketch of some other essential ingredients below, and leave a more detailed account to a separate paper.

First of all, we have proposed an algorithm to compute Eq. (9) for a well-behaved utility function U, as defined in Section 6. This algorithm, called the *module-matching algorithm*, is very intuitive. Roughly speaking, it looks for the *projection* of the commodity Y_i to $[U^l]$, i.e. a measure of distance between a commodity Y_i and the preference $[U^l]$.

Second, once the operational meaning of Eq. (9) becomes clear, it is possible to infer the *reservation price* which a consumer would like to pay for the commodity Y_i from $[U^l]$. Furthermore, given a set of commodities and a budget constraint, a notion of the *optimal choice* for a utility-optimizing consumer can also be developed. Up to this point, the rudiments of demand analysis are all there.

We then come to the supply side of the economy. The first question concerns the *cost function*, i.e. the counterpart of Eq. (9). Intuitively, the cost of a commodity should be related to the *complexity* of the commodity. The parse-tree representation of a commodity gives a natural measurement of it, i.e. the number of nodes in the respective parse tree, or in terms of GP, the *node complexity*. A simple cost function can be just a linear function of the node complexity. To make it delicate, we have to know more about the material markets and the knowledge markets (the processor markets).

Finally, the interplay between demand and supply is sufficient to initiate an agent-based simulation of the evolution of technology. An economy is initialized with a fixed number of consumers and producers. Consumers' preferences $[U^l]$ are exogenously randomly generated, and their endowments are decided in the same way. Producers are able to provide these consumers with a menu (a list of commodities) which are also initially randomly generated. Utility-maximizing consumers keep on searching for the commodities that can satisfy them most, and profit-maximizing producers keep on developing more suitable commodities. The process of searching for more suitable commodities can be then regarded as the first approximation of the evolution of technology. This population-based search process is driven by genetic programming which regards profits as the fitness function. The determination of profits crucially depends on the details of the price determination, which in turn depends on trading institutions and arrangements.

Needless to say, the computer simulation of the entire model is by no means trivial. We are now still in a trial-and-error process. Inevitably, we started with some very simple settings, for example, an economy composed of many consumers, but only one or few producers. We observe some very primitive simulation results on the evolution of technology and the accompanging amelioration of social welfare. However, the progress has not been smooth in the sense that quite severe fluctuations have been experienced. Drops in social welfare highlight a risk that the society may suffer from the evolution of technology as well. On the other hand, the progress may not be sustained long enough. The economy tends to stagnate after a short but fast take-off, and most consumers are only supplied with some "basic needs".

References

1. F. Allen and R. Karjalainen, Using genetic algorithms to find technical trading rules, *Journal of Financial Economics* **51**(2), 245 (1999).
2. M. Aoki, Open models of share markets with two dominant types of participants, *Journal of Economic Behavior and Organization* **49**, 199 (2002a).
3. M. Aoki, Applications of Ewens–Pitman–Zabell inductive methods in new economic dynamics, *Proceedings of the Sixth International Conference on Complex Systems*, 29 (2002b).
4. J. Banville, *Kepler: A Novel (Vintage International)* Vintage Books (1993).
5. G. Basalla, *The Evolution of Technology* (Cambridge University Press, 1988).
6. C. Ferreira, Gene expression programming: A new adaptive algorithm for solving problems, *Complex Systems* **13**(2), 87 (2001).

7. S. Levy, *Artificial Life: A Report from the Frontier Where Computers Meet Biology* (Vintage, New York, 1992).
8. C. Neely, P. Weller and R. Dittmar, Is technical analysis in the foreign exchange market profitable? A genetic programming approach, *Journal of Financial and Quantitative Analysis* **32**(4), 405 (1997).
9. A. Willihnganz, Software that writes software, *Salon. Com*, WWW Article (1999).
10. S. Zabell, Predicting the unpredictable, *Synthese* **90**, 205 (1992).