Agent-based Microsimulation of Economy from A Complexity Perspective

Yu CHEN, Xiaomeng SU, Jinghai RAO, Hui XIONG,
Renmin University Of China
E-mail: Yuchen@pubic.bta.net.cn or {mmsu, jhrao, huixiong}@263.net

Abstract

An often-noted feature of computational agents research is its interdisciplinary nature. The growing use of computational models in social, biological and economic systems raises many question and concerns. Platforms such as Swarm enable researchers to construct detailed, robust computational models. The availability of these platforms may speed the pace of the computational revolution and open new areas of interest. To test the usability and suitability of Swarm to model social-economy phenomena, we have made some simple economic models implemented in Swarm. First part of the paper may discuss briefly the advantage of agent simulation in modeling economic issues and why we chose Swarm as the platform. What followed is the description of the models and results from runs of the model are also reported. The last part may introduce some of our own insight in the realm of agent-based micro simulation model of the economy

Keywords

Complex Adaptive System (CAS), Swarm, Simulation of Economy, Genetic Algorithm Learning Classifier System (GALCS)

An often-noted feature of computational agents research is its interdisciplinary nature. The growing use of computational models in social, biological and economic systems raises many question and concerns. Platforms such as Swarm[1] enable researchers to construct detailed, robust computational models. The availability of these platforms may speed the pace of the computational revolution and open new areas of interest.[1] To test the usability and suitability of Swarm to model social-economy phenomena , we have made some simple economic models implemented in Swarm.. First part of the paper may discuss briefly the advantage of agent simulation in modeling economic issues and why we chose Swarm as the platform. What followed is the description of the models and results from runs of the model are also reported. The last part may introduce some of our own insight in the realm of agent-based micro simulation model of the economy.

We assume that readers of this paper have a rough idea on Swarm approach. See reference [2] first if otherwise.

1. Introduction

1.1 Economy as an evolving complex system

The canonical economic model is based on a "top down" view of markets or transactions [3]. For example, in General Equilibrium Theory, solutions depend on an omnipotent auctioneer that brings all production and consumption plans in the whole economy into agreement. Keynesian theory adopted the idea of an aggregate demand and supply function for the whole economy. Recent work in macroeconomics is often based on theories that model the entire economy as one agent. Problem is obvious with these approaches – they do not model the process which agents must go through to acquire information, evaluate choice and transaction.

A different approach to economic modeling is developed in recent years because of the limitations illustrated above. It has been described as "bottom up" or "process based." This approach is based on a new complexity perspective in economics. The economy can be viewed as an evolving complex system [4]. Six features of an economy should be point out from a

[1] Swarm is a library of software components developed by Santa Fe Institute, which allow researchers to construct discrete event simulations of complex systems with heterogeneous elements or agents. These libraries provide reusable objects for analyzing, displaying and controlling simulation experiments. Swarm is not based on any assumptions about the system that is being simulated, and is currently being used by a wide variety of researchers in the social, biological and culture sciences.

complexity adaptive perspective, which helps explain why the new methodology is superior to the older.

- **Dispersed Interaction** What happens in the economy is determined by the interaction of many dispersed, possibly heterogeneous, agents acting in parallel. The action of any given agent depends on the anticipated actions of a limited number of others agents and on the aggregated state these agents co-create.
- **No Global Controller** Computational models, like the real social worlds, do not have an active central planner. Instead, controls are provided by mechanism of competition and coordination between agents. Economic actions are mediated by legal institutions, assigned roles, and shifting associations. The amazing thing is that in this decentralization circumstance, coordination behavior emerged in the absence of a coordinator. Say, the order in markets.
- **Cross-cutting Hierarchical Organization** The economy has many levels of organization and interactions. People, household, castes, nationalities …Units at any given level typically serve as "building blocks" for constructing units at the next higher level. The overall organization is more than hierarchical, with many sorts of tangling interactions (association, channels of communication) across levels.
- **Continual Adaptation** The basic element in a complexity adaptive system is adaptive agent, whose behaviors, actions, strategies and products are revised continually as the individual agent accumulate experience. Thus the environment which is co-created by these heterogeneous adaptive agents constantly adapts.
- **Perpetual Novelty** Although an economic agent's ability to learn from one day to the next requires substantial regularity, its adaptive responses combined with exogenous shocks create a perpetually novel world. Thus, an accurate model of the world should not be teleological. We should no know where we are headed, just like the way "nature" solves problems Although there may be a tendency towards stable points (equilibria) if those equilibria keep moving, the novelty may be of greater predictive and explanatory value than equilibrium. (In contrast to General Equilibrium where economy vibrates slightly over the equilibrium point)
- **Out-of–Equilibrium Dynamics** Evolution need not lead to optimality, because in a perpetual novel world, the definition of "optimal" is hard to define. Global equilibrium is far from reaching, because of improvements are always possible and indeed occur regularly. Again "evolution" is the way we can simulate "nature" does.

These properties are by no means excluded with economics. Systems with these features have come to be called "adaptive nonlinear networks" [2] Other than economies, there are many such systems in nature and society, like nervous systems, immune systems, ecologies. These features, which have no place in canonical economic models, require a new classes of combinatorial mathematics and population-level stochastic process, in conjunction with computer modeling. These computational techniques are in their infancy. They emphasize the discovery of structure and the processes through which structure emerges across different levels of organization. That is, they focus on how structures (orders) emerge from bottom up and why they happen. A key focus of the research is understanding how global regularity arise from the bottom up, through repeated local interactions of autonomous agents channeled through socio-economic institutions, rather than from top down coordination mechanisms such as imposed market clearing constraints or an assumption of single representative agents [7].

As we can see from the above illustration, the complexity perspective of economics is more aligned with the reality. Thus models based on this theory are more realistic. Microsimulation of the economy refers to a model that simulates the actions of economic decision-makers individually and then generates macroeconomic quantities of interest by integrating those actions. The building blocks of these models are heterogeneous agents, which represents real life economic decision-makers, such as household, firms or government. Aggregates of the agents' microeconomic actions generate macroeconomic quantities of interest In a complexity point of view, that is we model agent-level interactions and try to find what may happen in aggregated level. This approach affords several advantages over traditional techniques (macroeconometric or computable general equilibrium [CGE]) [5].

- The procedure does not require a functional form for its endogenous relationships (as

[2] John Holland's outline at the 1987 meeting beautifully and presciently frames these features. For an early description of the approach , see also the program's 1989 newsletter "Emergent Structures"

macroeconometric and CGE models do). The user has greater freedom when modeling individual agent behavior and can model in detail. For example, the learning of some agents by using GA (genetic algorithm). In addition, the effect of certain nonlinear legal, regulatory or policy changes can be modeled explicitly.

- The approach emphasis on disequilibrium growth paths. Its analysis is based on evolution and emergent behavior rather than on a mechanistic view of society, and it uses learning algorithms to simulate the behavior of some agents rather than an assumption of perfect rationality.
- The procedure is individual-agent-based. Therefore, the user must build microeconomic models of the individual decision-makers rather than macroeconomic models of markets. Existing rich sources of micro-level data are available.
- In contrast to most models in economics, that place emphasis on *explaining* a fully grown socially observed phenomenon, another possible way to understand problem is to emphasis on *growing* the phenomenon rather than explaining it directly.[6] This is consistent with the view that economy is an evolving system.
- The user can easily introduce a stochastic element using a simple random number generator. It becomes much easier with Swarm, which provides numerous random number generators (from uniform distribution to exponential distribution).

1.2 The Swarm Platform

Computational models of social and economics systems is by no means new, of course,. What's new is that the system: is individual agent-based, has hierarchy, is based on repeated local interactions of agents and most important both the individual agent and the system as a whole are evolving through adaptation. Till now, the theory has told us how to shoot, and what is needed is a right gun. Swarm is just the right weapon. We use Swarm as a platform for our novel research on economics as an evolving complex system for the following reasons.

- Swarm is object-oriented. Agents in the model represent real-life decision-makers like household, firms and can be easily mapped into an object in object-oriented programming. As illustrated in figure 1

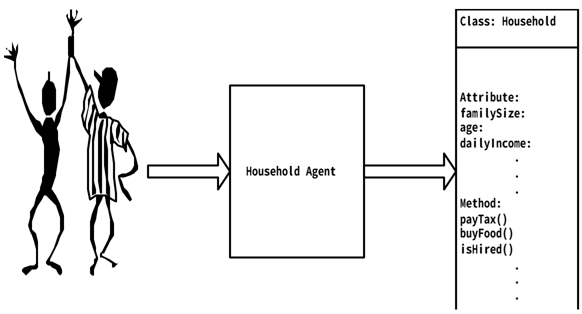

Figure 1 Agent and its Object implementation

- Swarm is developed by (Santa Fe Institute) SFI, the birthplace and research center of Complexity Adaptive System (CAS) theory. Thus it is more aligned with the theory.
- It is GNU software, with all its advantages and disadvantages for being a GNU. Its first release was in 1995 implemented in Objective-C. The Java version Swarm was released in 1999. There is a friendly user group and support center for Swarm (www.swarm.org), which is very important for experience sharing and problem solving.
- Swarm is problem domain unrelated. It is not based on any assumptions about the system that is being simulated, and is currently being used by a wide variety of researchers in social, economics and biological science. Swarm intends to help researchers focus on theoretical model rather than on tool building by giving them a standardized suite of software tools that provide a well equipped software laboratory[8]

Every coin has two sides. Swarm is no exception, of course, in particular given the learning barrier for Objective C. But compared with constructing our own home-grown computational experimental tools, most of which from a software engineering perspective are poorly designed. Swarm, as it stated, an efficient, reliable, reusable software apparatus for experimentation, is a big step forward.

Well little is really worthy before we try it ourselves. To test the usability and suitability of Swarm to model social-economy phenomena, we have made some simple economic models implemented in Swarm. One of them is a simple market economy model, which will be specified next. If the model is success, it will serve as a prototype for a macroeconomy framework model.

2. A simple market economy model.

2.1 Description

The prototype of our first model is a simple market economy (product market and labor market). The model contains three kinds of economic agents: households, firms that produce food, and government. We use genetic algorithm to simulate the learning process of some agents. Figure 2 is the main interaction diagram of the agents. For each agent, most of their actions are involved either with collecting money or spending money.

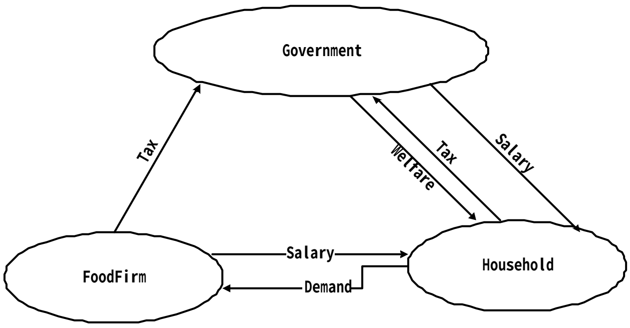

Figure 2 Simple Interaction Diagram

2.2 Agents

2.2.1 Household

Income:

Household agents represent their real-life counterparts. They generate most of their income through employment. Employers can be one of the firms or the government. For simplicity, only one of the family member can be hired. Once the household agent is hired, he receives salary from the employer every day until he is fired. Government will allocate welfare to those unemployed, the payment amount is a function of family size. Other incomes the households may get include shares of the company profits.

Consume:

- The household pays a flat-rate income tax on salary.
- Buy food. The household consumes food every day. Demand for food is based on family size. Every day, when this demand is determined, the household agent chose a suitable food firm. The decision on where to buy food is made by the following process.

1. The agent first consults a list of food price (each food firm broadcasts its price message everyday).
2. If firm A offers food for price p(A), the household will buy food from this firm with a probability possibility(A) = $p(A)^{-q}$ (the possibility(A) will be normalized)
   Where q is a given exogenous demand parameter
3. After calculation possibility() for each firm, a vector is generated in the form of [possibility(A1),possibility(A2), …..possibility(An)] (suppose there are n firms) The higher the possibility(A) is , the greater chance the household will have to buy food at firm A. In short, this means from a firm's perspective the comparatively lower price p(A) is in relation with a great chance of being demanded

2.1.2 Firm

The firm's income comes from its sale of product.
Profit = price*salesAmount – productionCost
The changing of price or production may affect the sales amount (because consumers will make their food consume decisions according to the price message the firms offered) and ultimately change the profit of a firm.

Production

There is only one type of firm in current version – food firms. The firms have a production function production = c* K^a *L^b

Where

K represents the number of machines on hand in the firm.

L represents the number of employees.

a,b,c are constants that help to adjust production .

From the above formation, we can see that a firm can change its production by varying K or L. The current version sets K to a constant. Future version will allow the firm to take a loan to buy new machines every year. As to L, the firm may hire or fire workers every day. The decision of employment is based on the recent average daily demand and the current inventory level.

How to determine product price

The model uses Genetic Algorithm Learning Classifier System (GALCS) [5] to simulate a firm's price decision process. In the end of every day, when all sales have been done, a firm may calculate its sales amount and profit. According to today's information and the history information a firm may answer four question (I) Whether profit has been increasing recently (II) Whether sales have been recently increasing or not (III) Whether product price has been increasing or not (IV) Whether prices are higher than the industry average or not. We use a four-element vector to hold the result. If the answer is yes we assign 1 to this sector and 0 if otherwise.. Thus a vector (1,0,1,0) means profit has been increasing today; sales have been decreasing compared with recent history average; product price has been increasing and price is lower than the industry average.

According to the result of answers to (I) through (IV), a firm may find itself in one of the 16 states which are (0,0,0,0) (0,0,0,1) (0,0,1,1) …..(1,1,1,1)
The GALCS assigns a probability vector (p^D, p^I, p^C) to each state,

Where p^D = the probability that the firm will decrease the price the next time the firm enters the same state.
p^I = the probability the family will increase the price
p^C = the probability the family will keep the price unchanged.

Upon entering a certain state, the firm agent makes decision by using the corresponding probability vector and choosing a random number. The agent then adjusts the probability vector according to the profits acquired under the chosen decision. Here, a firm agent's fitness function is its profit.

For instance, suppose at a particular time, after answering the four evaluating questions, the firm enters the state (1,0,1,0), and the corresponding probability vector (p^D, p^I, p^C) = (.2, .5, .3), which means the possibility for increasing price is higher. Suppose the firm draws a random number of 0.25446543654353, which indicates the need for increasing price. Suppose further that as a result of increasing price, profits drop (because of a higher price in market is less competitive), which is against the decision of raising price at this state, indicating the fitness of this solution is decreasing. The vector is then adjusted to (.25, 4, .35), means the raising price probability is lower next time re-entry this state. Thus, this model simulates the firm agent's learning process. The changed probability vector reflects the unlikelihood that the agent will raising price upon re-entry into state (1,0,1,0). By varying the price of the product, firms compete with one another. It is, in essence, similar to the result that game theory dictated. Firm agents can adapt their behavior (price strategy) dynamically, according to the changing market environment which they co-create and their past experience. Thus, both the agent and the whole environment will evolve through learning

Disbursal of profit

A firm pays tax to government in relation with its profit. In this version, the after-tax profits is distributed equally to all households in the entire economy, which is consistent with the spirit of general equilibrium [10] [3]

2.1.3 Government

The government's function is to collect taxes from firms and household, as well as pay assistant welfare to the unemployed.

3. Results

Actions of each agent are arranged in a time schedule according to occurring sequence. Every day, the time schedule will be executed once. Certain results will emerge after some time of runs. With the help of Swarm, we can easily observe the continuous changing status of any given factors. The output of data in Swarm is mainly handled in two ways: either by graphical tools or with prints and file commands that can be saved for later statistical use.

We implemented the model with Objective C and Swarm. The model runs on Sun Ultra Enterprise 3000 (Solaris 2.5.1) Some of the results are reported below. Figure 4 is an example of a run using the simple model for about 20000 days. We can easily adjust some of the initial parameters by the input window of this model. Figure 3 is the initial parameter input window. Among the values, "numHousehold" is the number of household agents; "numFirms" is the number of food firm agents, "nQparam" is the q parameters in the household's food demand function, which can affect the price sensitivity of firms. In the situation figure 3 represents, the values are 1000 household agents, 4 food firm agents, 1 government.

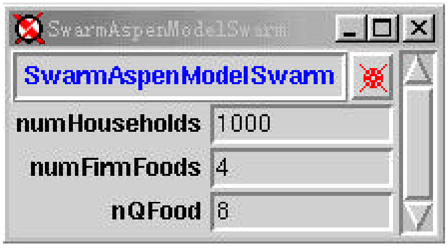

Figure 3 Initial value input window

[3] Of course, there should be a delay in the disbursal to the whole economy. Currently, a firm will distribute its after-tax profit in the end of every month

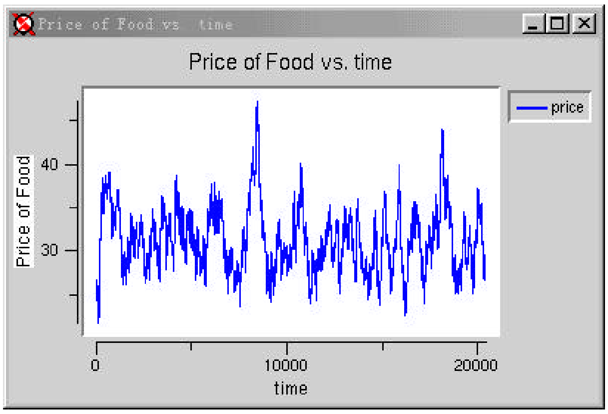

Figure 4 Average industry price in the runs of 20000 days.

As we have stated in the previous paragraphs, through simulating individual agent level interactions, we can observer some aggregated level structures emerge from bottom up. In figure 4, an obvious business-like cycle can be observed. This cycle can be explained as result from pricing decisions of the firms, their decision on hiring or firing workers and inventory excess. Figure 5 is the aggregated level situation in the time cycle of 1000 days. Unlike figure 4, there is no obvious cycle in this time. We explain it as: the adaptation process is not mature in such a short time and the firms are still in their early stage of learning.

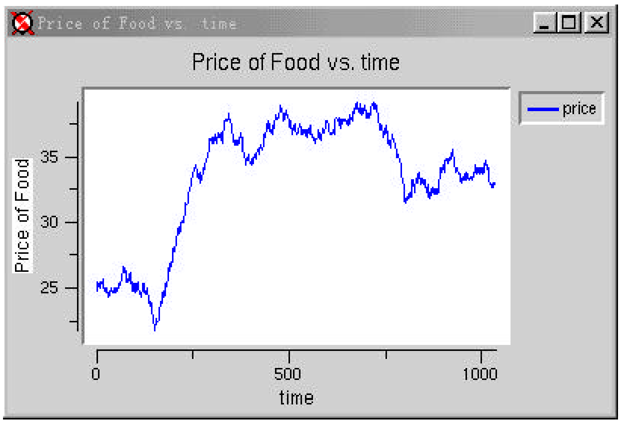

Figure 5 Average price in the runs of 1000 days.

By adjusting the q parameter, we can change the firms' sensitivity to price variation. A higher q parameter indicates a higher sensitivity to price change. Figure 6 is a price sensitive market where the q parameter is 15.0 and figure 7 is a relatively price non-sensitive market with q equals 8.0 As shown is figure 6 and figure 7, a price sensitive market has a relatively shorter period of cycles. A two-person game may be described where the players are firms and the strategies are the prices the firms charge. Under steady-state demand conditions, a unique equilibrium of the game occurs where p1=p2=qC/(q-2)[4]. In this prototype, prices hovered near these values. Here, the genetic algorithms lead to prices dictated by theory.

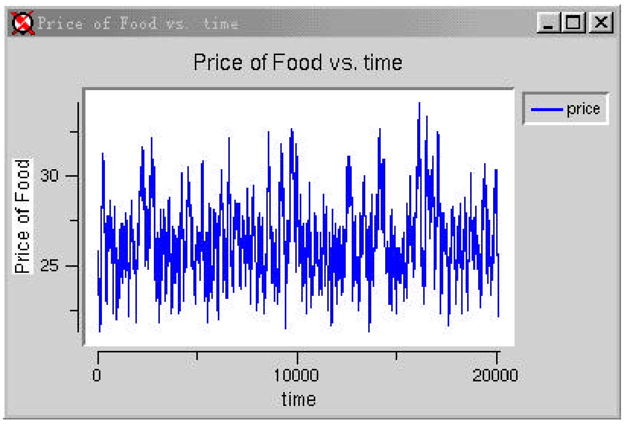

Figure 6 a price sensitive market

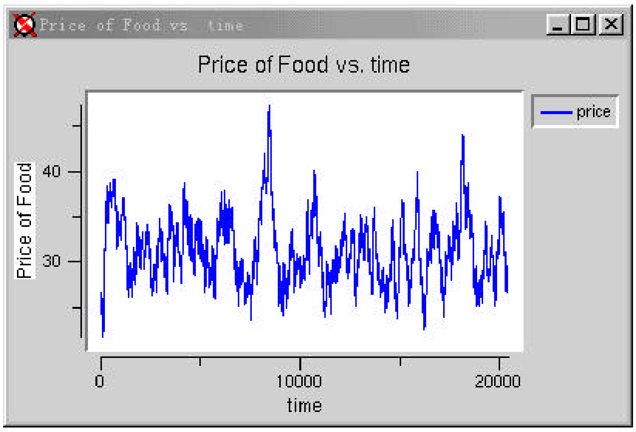

Figure 7 a relative price non-sensitive market

4. Conclusion and future extensions.

These simulation results suggest that Swarm is a usable platform for our micro-simulation modeling. The agent learning process is proceeding in the right directions. Although the genetic algorithm we use here is comparatively simple and lack of potential to new knowledge discovery, its clear and neat logic as well as the basic learning ability, qualifies the algorithm a useful candidate for simple models. It helps to set up the prototype quickly and can demonstrate whether the modeling is in the right direction. After the prototype is well done, we can substitute the agent's learning strategy with a cleverer

[4] C is the cost of each piece of goods, whose value is 22.5 in this model.

one. The problem, how to make the agent rational is a general AI problem, which beyond the scope of this paper.

Our experience in building a number of economic models using Swarm proves that such models can be developed quickly and their performance are satisfactory. How to evaluate the way we use computational agent to simulate economic systems is still yet to be determined. But, at least, this is a way that we can use to study and explain the emergence of orders in a decentralized market in the absence of coordinator.

A more detailed model has been designed and is now under implementation. This model includes households, banks, government, China center bank which issues bonds and various types of firms. We intended to use the model as a framework for macro-economy research.

5. Acknowledgement.

The project is funded by China National Science Foundation. The authors would like to thank Mr. Marcus G. Daniels (Swarm developing Group, SFI) for his help in the process of installing Swarm on Windows ™and Solaris ™systems. Many thanks also to Dr. Wen Jirong, who helps download many useful relevant materials for us.

REFERENCE

[1] Scott E. Page (1999) *Computational Models from A to Z*

[2] Nelson Minar, Roger Burkhart, Chris Langton, Manor Askenazi. *Swarm Documentation* [online] http://www.santafe.edu/projects/swarm

[3] Benedikt Stefansson (1997) *Swarm: An Object Oriented Simulation Platform Applied to Markets and Organizations.* Evolutionary Programming VI Springer-Verlag, New York, 1997

[4] W.B.Arthur, Steven Durlauf, David Lane (1997) (Eds.). *The Economy as an Evolving Complex System II* Addison-Wesley, Reading, Mass., 1997

[5] N.Basu, R.J.Pryor, T.Quint and T. Arnold (1996) *ASPEN: A Microsimulation model of the economy,* Sandia report.. SAND96-2459

[6] N.Basu and R.J.Pryor (1997) *Growing a Market Economy* Sandia report. SAND-97-2093

[7] Leigh Tesfatsion *Teaching Agent-based Computational Economics to Graduat Students* [online] http://www.econ.iastate.edu/tesfatsi/

[8] Nelson Minar, R.Burkhart ,C.Lanton and M.Askenazi (1996) *The SwarmSimulation System: A Toolkit for Building Multi-Agent Simulations* SFI working paper [online] http://www.santafe.edu/projects/swarm

[9] Xavier Vila, *Adaptive Artificial Agents Play a Finitely Repeated Discrete Principal-Agent Game* R.Conte, R.Hegselmann, P.Terna Eds. *Simulating Social Phenomena* Springer 1997

[10] Varian,H., *Microeconomic Analysis*, New York: W.W.Norton and Co, 1978

[11] Tracy Mullen , Michael P.Wellman, *Some Isues in the Design of Market-Oriented Agents* [online] http://www.si.umich.edu/~prie/

[12] John H. Millar *Evolving Information Processing Organizations* SFI Working paper. 95-06-53, 1996

[13] Zhongzhi S. (Eds.) *Advanced Artificial Intelligence* China Science Press(1998)

6. Biography

Yu Chen is a Professor of Computer Information Systems and the Director of Economic Science Laboratory at Renmin University of China. Among the ten books he authored or co-authored are "Introduction to Economics of Information Management", "A Dialogue about Systems" and "Information Economics". He authored over 50 articles that appeared in journals in China and the U.S. He has been an educator his entire career that includes consulting projects with major Chinese SOEs. His research areas are information economics, system science and management information systems.

Xiaomeng Su and Jinghai Rao are currently graduate students at the Information School of Renmin University. They have been the Swarm Programmers and the major developers of Economic Simulation. Their research interests include: analysis and design of MIS, Electronic Commerce and Complexity Adaptive System.

Hui Xiong now is a doctoral candidate of the Business Management School of Renmin University of China. His research interests major in CAS theory, computer simulation on economics and Management Information System. He is one of the designers of the economic models and also a developer of the models. In 1994, he graduated from the Information School of

Remin University of China and gained bachelor degree, then he continued his graduate career in this school and in 1997 obtained master degree.