Modeling and Analysis of Agent-based Artificial Demand-Supply Market by Using the Genetic Programming and its Applications

時永, 祥三
九州大学大学院経済学研究院

https://doi.org/10.15017/3774

出版情報：經濟學研究. 71 (2/3), pp.107-118, 2005-03-25. 九州大学経済学会

Modeling and Analysis of Agent-based Artificial Demand-Supply Market by Using the Genetic Programming and its Applications

Shozo Tokinaga

September 30, 2004

1 Introduction

Over the past few years, the agent-based computational systems have gradually become one of major tools in the analysis of socio-economic dynamic systems (Lebaron 2002, Tesfasion 2000). Several models of artificial financial market such as stock market based on interacting heterogeneous and autonomous agents have been developed to cope with the mathematically untractable problems (Tesfasion 2000).

In the same way, there are literatures on the control of the input pricing in the service facility where the heterogeneous self-optimizing agents (customers) are assumed to join queueing system and the input pricing reveals as a chaotic time series (Stidham 1992, Rump.et al. 1998).

In this paper, we consider the dynamics behavior on the input pricing mechanism for artificial market where the members (called as agents) are regarded as agents allowed to learn from their past experiences (Holland et al. 1992). Agents decide to join/balk to the market by predicting the future demand (the prediction corresponds to the input pricing in the service facility) to obtain the profit. If the agent estimates the pricing is appropriate and the profit is expected, then the agent supplies the goods into the market. Otherwise (the pricing is not satisfactory), the agent decide to balk (not join) into the market. The most important problem in the input pricing is the feature of the pricing time series which is the result of self-optimizing agents' behavior. Sometime the pricing time series fluctuates very rapidly and becomes to be unstable, and sometime shows convergence to a certain but insignificant level. Then, the feature analysis of the pricing time series is important to the study of market structure.

To emulate more realistic environment in markets, agents are assumed to be heterogeneous and to have their own rule for predicting the future pricing. Their learning processes are modeled by the co-evolutionary Genetic Programming (GP) which is a familiar tool for modeling the multi-agent systems (Chen et al. 2002,2003, Ikeda. et al. 2001, 2002,2003, Koza 1990, 1991,1992). Then, five types of agents are assumed in which agents with random behavior are included. Two types of market are assumed depending on the demand and supply in the market. In type I market, the input pricing is determined instantaneously by assessing the demand/supply of the agents. On the other hand in Type II market, the supply to the market acts as an inventory prepared for the demand, and the accumulated amount of the good is estimated by the agents. In the simulation study, we examine the condition for the system parameter to observe the chaoticity in the time series of input pricing.

Due to the limitation of the capacity in service facility and the self-optimizing behavior of agents for the join/balk decision, the input pricing becomes chaotic time series under a certain condition for parameters. In the multi-agent system, it is necessary to show the effect of the ratio of each type of agent on the chaoticity of input pricings as well as the parameter of market. Then, by changing the ratio of the number of agents, the pricing time series bears the fractality. The fact imply us the ability of the multi-agent system to approximate the real time series of input pricing. .

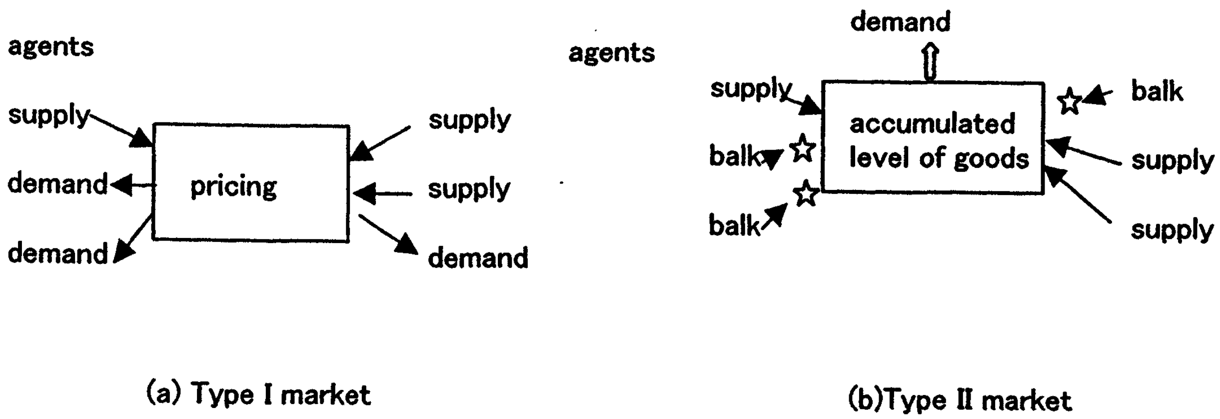

Fig.1 Two types of demand-supply markets.

2 Types of agents

2.1 Agents of type 1 and type 3

The major component of this architecture is the heterogeneous adaptive agents. Specifically, five types (three categories) of agents are defined in this architecture. The agents of both type 1 and type 3 are agents who forecast the value of input pricing of the next period by using an adequate forecast equation model selected from a forecast model base. The difference of these two types is that the agents of type 1 possess their individual forecast model bases, but the agents of type 3 only learn from a public forecast model base, without their own bases. This public forecast model base, which can be supposed like a mass media providing a public place for social learning, can be accessed by all agents of type 3 simultaneously.

Besides the agents of type 3, the agents of type 1 may also make a decision to access this public forecast model base when they feel unsatisfied with the growth speed of their wealth and all the equation models they own seem not effective enough. But compared with learning from public forecast model base, they prefer updating their own forecast model bases more frequently when necessary, because they perhaps have more confidence with own equation models. Therefore, a so-called stochastic learning mechanism is presented in our model, letting the agents select from these two alternatives stochastically.

In usual decision making, agents use best prediction or rules which promise us the best prediction of pricing based on the past record of price. However, if the environment of the market is fluctuating, this kind of deterministic scheme lead agents frequent change of trades, and it results in poor strategies. Therefore, the conventional method of LCS (Learning Classifier System) is employed in which agents select prediction or rule by activating appropriate clusters promising relatively better results (Holland 1992). .

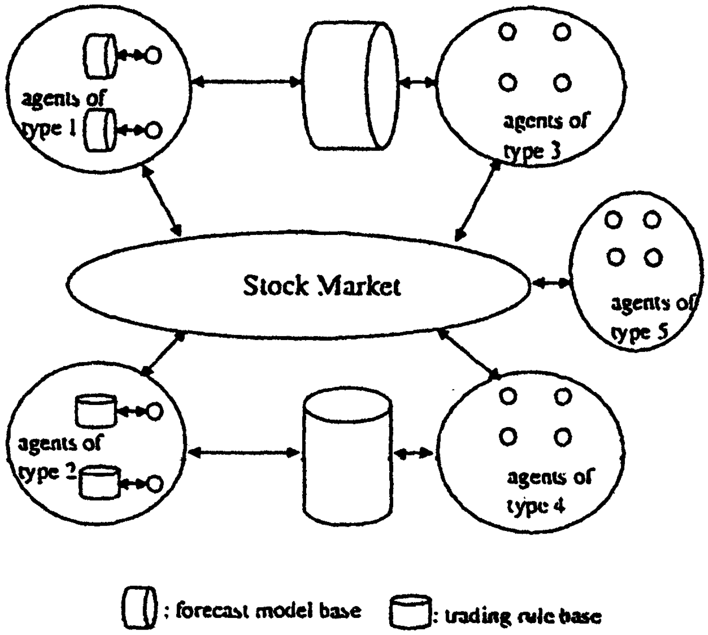

Fig.2-Overview of artificial market and multi-agent system

2.2 Agents of type 2 and type 4

Moreover, in reality the complexity of the market forces agents to act inductively, using simple rules of thumb. Using these simple trading rules to make a supply decision, seems to be effective at some time. For example, they use production rule to decide join/balk (supply or not supply) goods in the next time point. Based on this consideration, then we design the agents of type 2 and type 4, who will use production rules to support their decision. The difference of these two types is that the agents of type 2 possess their individual trading rule bases, but the agents of type 4 only learn from a public trading rule base, without their own bases.

2.3 Agents of type 5

Even though the agents of type 1, type 2, type 3 and type 4 have different characteristics respectively, they do have one common characteristic, namely their rationality when making a decision whether to supply or not. On the other hand, different from these agents above, the agents of type 5 seem to behave irrationally, in the way that they do not use any reasonable approaches to support their decision making process.

3 Learnign based on theGP

3.1 Agents (Type 1 and 3) using Prediction Formula based on GP

Since the GP procedure are applied in various fields and many results are available, we omit the details of the GP in the following (Chen et al. 2002,2003, Ikeda.et al. 2001, 2002,2003, Koza 1990, 1991,1992) . In the GP, each forecast model is represented in the tree structure (called individual).

In the parse tree, the non-terminal node is taken from the function sets, containing followings. $+, -, \times, \exp, abs, sqrt, \log, \min, \max, av, price$. The explanations about the functions like

min, max, $av, price, dividend$, having only one operand, are shown as follows, in which time t denotes the current period. Terminal node consists of arguments chosen from set of constants.
$\min(t)$: minimum pricing in period $[t-1, t]$
$\max(t)$: maximum pricing in period $[t-1, t]$
av(t) : average pricing in period $[t-1, t]$
price(t) : pricing in period t

We iteratively perform the following steps until the termination criterion has been satisfied.

(Step 1)

Generate an initial population of random composition of possible functions and terminals for the problem at hand. The random tree must be syntactically correct program.

(Step 2)

Execute each individual (evaluation of system equation) in population by applying the optimization of the constants included in the individual. Then, assign it a fitness value giving partial credit for getting close to the correct output.

(Step 3)

Select a pair of individuals chosen with a probability π based on the fitness. The probability pi is defined for ith individual by using the fitness S_i as follows.

$$\pi = (S_i - S_{min}) / \sum_{i=1}^{N} (S_i - S_{min}) \tag{1}$$

where S_{min} is the minimum value of S_i and N is the population size. (Step 4) Then, create new individuals (offsprings) from the selected pair by genetically recombining randomly chosen parts of two existing individuals using the crossover operation applied at a randomly chosen crossover point. To introduce the diversification in the individuals, the mutation operation is allied to a randomly selected individuals at a certain probability by replacing a symbol by another symbol.

(Step 5) If the result designation is obtained by the GP (the maximum value of the fitness become larger than the prescribed value), then terminate the algorithm, otherwise go to Step 2.

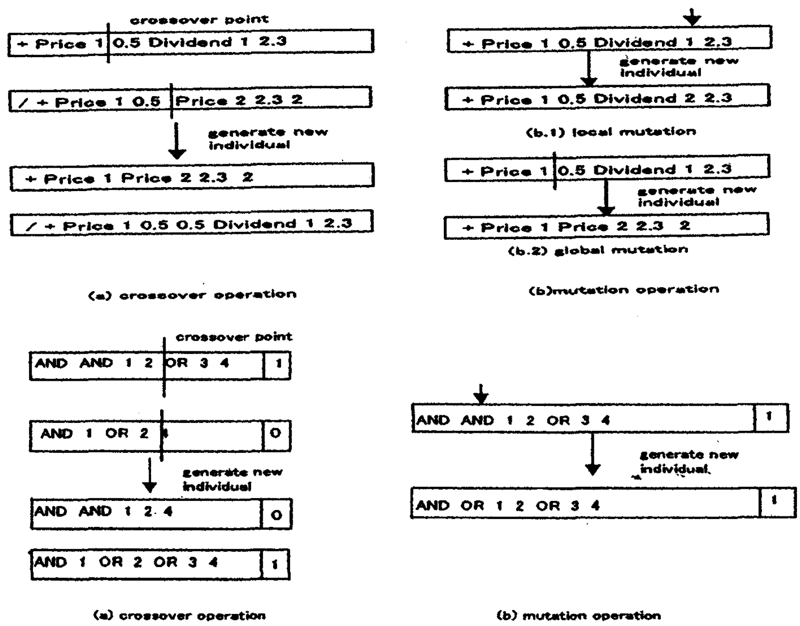

Fig.3 Scheme of the Genetic Programming

We define the fitness as the accuracy of forecast equation model on historical datum. Specifically, the fitness of forecast model is the reciprocal number of squared forecast error denoted as $e_{ij}(t)$ in which i is the index of agent, and j is the index of forecast model in the base and t accounts for time horizon. The value of $e_{ij}(t)$ can be calculated in a smooth exponentially weighted fashion as follows in which w denotes the weight.

$$e_{ij}(t)^2 = (1-w)e_{ij}(t-1)^2 2 + w[(P_t - E[P_t])]^2 \tag{2}$$

In this paper, we let the variance $\sigma(t)^2$ estimated by agents of type 1 and type 3 equal to the squared forecast error of selected forecast model.

At the end of every period, the fitness of all forecast models in agent's individual forecast model base and public forecast model base will be reevaluated automatically according to their performance. Agents (Type 2 and 4) using Rules based on GP These rules are called condition-action rules, which means if the condition part of the rule is satisfied then the action represented by the action part will be implemented. Moreover, these types of agents are designed to supply or not supply goods at a stochastic quantity.

We can define the condition part of a rule as a logical expression which is represented as a connection of statements with logical operators, including AND, OR, NOT. And the statement can be defined as connection of two arithmetic expressions (equations) with comparative operators, including $>, \geq, <, \leq$. The arithmetic expressions in statements can be generated in the same way as described in previous section.

Similarly, a logical expression is also represented as an individual in the GP. More precisely, The condition part is also represented in the tree structure and the same style of prefix representation. To simplify the system configuration, we assume that the condition part of a rule consists of logical expressions which are represented by a single logical operator and two statements (a combination of binomial logical expressions). Then, the condition part can be represented by a prefix representation like an arithmetic expression where the arithmetic operators are replaced by logical operators. Each statement in logical expression can be represented by two arithmetic expressions and one comparative operator.

The relation between the logical expression and statements are realized by a hierarchical data structure with two level of pool of individuals. The overall expression of logical expression is stored in the first level, and the links to combine the statement stored in the second level are used to aggregate comprehensive data structure. The crossover and mutation operations for the logical expression are the same as the GP in arithmetic expression by replacing the operators and operands suitable for logical expressions.

But, to improve the fitness of individuals in the arithmetic expressions (individuals in second level pool of individuals), we apply the GP operations also to the arithmetic expression, at a prescribed probability.

4 Two Models of market

4.1 Type I market (Instantaneous supply)

In Type I market such as stock market the pricing can be found by balancing the demand and the supply of goods (stocks). For simplicity we also utilize the same price adjustment schema as used in conventional works.

The pricing can be adjusted according to the following equation, in which $Gamma(x)$ is a function of the difference between total quantity agents would like to buy denoted as B_t

and total quantity agents would like to sell denoted as O_t. Then we consider one form of function$\Gamma(x)$ as follows.

$$\Gamma(B_t - O_t) = \tanh(\beta(B_t - O_t))) \tag{3}$$

where the function $\Gamma(B_t - O_t)$ means the transformation function of demand into pricing. The function has the same parameter β either depending on the cases either the demand is greater than the supply or not.

4.2 Type II market (Accumulated supply)

We consider the dynamic behavior of an input-pricing mechanism for a market in which heterogeneous self-optimizing customers base their future join/balk (supply or not supply) decisions on their previous experiences (Chen et.al 2003). Each agent joins the market if and only if its value exceeds the estimated admission price(pricing).

Define the supply rate of goods as λ. For a given rate λ of agent, the accumulated amount of good is evaluated based on the market structure as $G_\mu(\lambda)$, where μ is the measure of demand capacity such as the needs for good. Define the admission price by

$$p = f + G_\mu(\lambda) = f + hW(\lambda) \tag{4}$$

where the parameter f means a fixed cost, and the parameter h means the unit cost for accumulated amount of good in the market and the function $W(\lambda)$ is ordinary system function used for representing the mean waiting time on the service facility under the assumption that the input rate is λ.

According to conventional queueing theory, we can easily obtain the functional form of $W(\lambda)$ represented by λ and μ. A potential arriving agent seeking to maximize its net benefit, has an incentive to join the system if its expected profit exceeds the admission price. Agents cannot observe the amount of good (pricing) in the market before deciding whether to join. Hence, they do not know the admission price, and use predicted price p_t.

Let λ_t denote the supply rate during the period t which induces a price $p_t = f + G_\mu(\lambda_t)$. Observing the price pt, the customer then collectively form a prediction price π_{t+1} for the next period by the following exponentially smoothing equation.

$$\pi_{t+1} = (1-\omega)\pi_t + \omega p_t \tag{5}$$

where $0 < \omega < 1$. Since each successive forecast π_t seeks to predict the pricing p_{t+1}, we can view this dynamic pricing process as an equilibrium seeking pricing algorithm governed by the first order nonlinear differential equation.

$$\lambda_t = F(\pi_t), p_t = f + G(F(\pi_t)) \tag{6}$$

$$\lambda = F(\pi) = \begin{cases} 1 \quad (0 \leq \pi \leq d); \\ (a-\pi)/(a-d) \quad (d \leq \pi \leq a); \\ 0 \quad (\pi \geq a) \end{cases} \tag{7}$$

The function$\lambda_t = F(\pi_t)$ means the (real) supply rate which represents the average number of goods actually entering the system per unit of time.

Clearly, if the arrival rate is lower than the capacity μ, of the facility, then the equilibrium is stable for all μ. However, if the capacity μ decreases below a sufficient large value, a period-doubling bifurcation of the equilibrium occurs. In the region, the process alternates between a stable two-cycle. If the capacity is decreased further, a period-doubling cascade then ensures, then in turn splits into a four-cycle, which then in turn splits into eight-cycle. Then, the band of Li-Yorke chaos begins as the service rate parameter is decreased (Li et al. 1975). Now, we extend the model of input pricing in service facilities to the multi-agent systems.

By the extension, we emulate various kinds of performances in the service facilities rather than supposing a single agent. We assume that the agents arrive to the market are not identical and they use several types of decision making rules. However, the extension is not so complicated compared to the single agent case, and it is straightforward. Following points are the difference of multi-agent systems.

(1) determination of supply rate

We assume that each agent i considers that all of the other agents in the system behave like him. More precisely, the agent i predicts the pricing at the next time point t+1 by using equation (5), and is represented as π^i_{t+1}. Then, the agent predicts the supply rate based on π^i_{t+1} by using equation (5)(6)(7) as follows.

$$\lambda^i_{t+1} = F(\pi^i_{t+1}), \pi^i_{t+1} = (1-\omega)\pi^i_t + \omega p_t \tag{8}$$

The agent i estimates the supply rate to the facility by multiplying λ^i_{t+1} by the number of whole agents N.

(2) accumulated amount of goods estimated by the agent i

The agent i estimates the functional form of mean amount of goods represented in equation (4) for the market by using the past observed data of pricing. The estimation is done by using the GP, and is denoted as $W_i(\lambda)$. At the same time, the agent i predict the pricing at the next time point $t+1$ as follows.

$$p^i_{t+1} = f + hW_i(\lambda^i_{t+1}) \tag{9}$$

Based on difference between the prediction p^i_{t+1} and the pricing p_{t+1} actually realized by the market, the agent examines the capability of prediction and then adjusts it by using the GP. Each agent improves the prediction of pt+1i based on the difference between the real pricing p_{t+1} and his estimation pt+1i using the GP.

(3) supply rate to the system

Since the agent i supplies goods to the market at the rate λ^i_{t+1} , then the aggregated value of them is the supply rate to the market.

$$\lambda_{t+1} = \sum_{i=1}^{N} \lambda^i_{t+1}/N \tag{10}$$

By using the total supply rate, the pricing in the facility is calculated as $p_{t+1} = f + hW(\lambda_{t+1})$. Each agent estimates the function of accumulated level time $W_i(\lambda)$ where the variable for function is only the supply rate λ on the basis of observation of pricing. Each individual in the pool corresponds to an approximation for $W_i(\lambda)$, then the difference between the observed pricing and the prediction calculated by the individual makes the fitness of individual.

5 Simulation results for chaoticity analysis

5.1 Type I market

In the following, we examine the behavior of multi-agent systems for approximating the pricing model in markets based on the simulation studies. At first, we consider Type I market, and we assume that the structure of the market (the function $B(B_t - O_t)$) is given by a know function, and the agents' behavior are characterized by the approximation of pricing. As is known, to prove the chaoticity of a time series, we must depict the attractor or equivalent return map by using the observed time series by showing a distinct plot of points.

Figure 5 shows an example of artificially generated stock price where the statistical data for the time series is very resemble to real stock prices. .

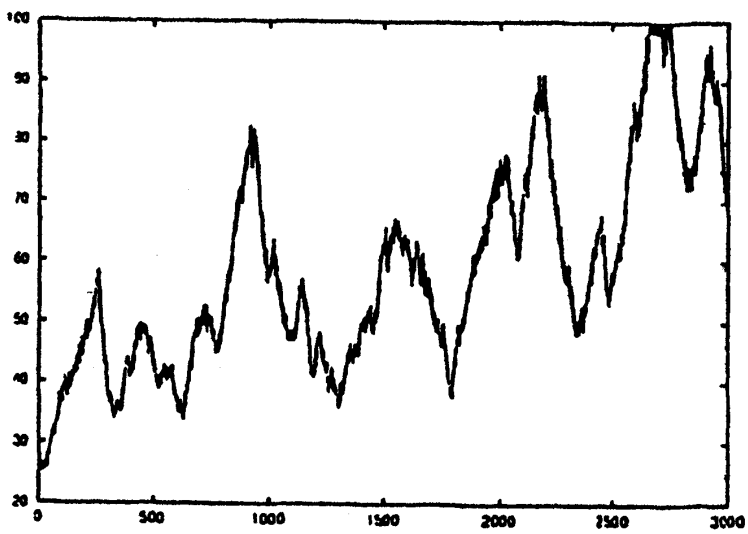

Fig.4-Example of artificial stock price

However, in the original configuration of our system, if we add agents of type 5 the pricing time series bears a kind of randomness due to the random behavior of type 5 agents. As a result, it is hard to show a distinct plot on a plane of attractor.

Moreover, the LCS adapted to fluctuating environment also bring a kind of randomness to the time series, since under the LCS agents select a relatively good prediction result at a random fashion using the pool of individuals. By considering these reasons, we postulate following three assumptions to prove the chaoticity of the pricing time series.

(1) type 5 agents

Agnets of type 5 are removed from the system.

(2) individual used by agents

Each agent used the individual having highest fitness for the prediction of pricing.

(3) co-evolutionary GP

Agents of type 3 and 4 use the common knowledge base. Then, the common knowledge base is assumed to be composed of the individuals which are composed of individuals with highest fitness in the pool of agents of type 1 and 2.

Table 1 shows the Maximum Lyapunov exponent (mLE) of input pricing. We see from the table that the mLE is positive and the pricing time series becomes to be chaotic under the condition $0.000037 < \beta < 0.0061$.

Figure 5 show the bifurcation diagram for the parameter β. .

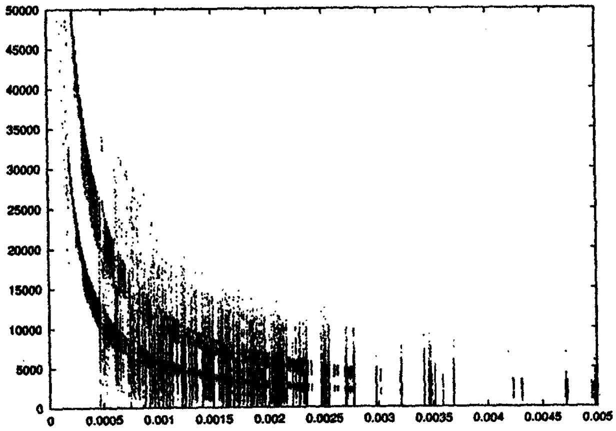

Fig.5-Bifurcation diagram dependding on β

Table 1. Maximum Lyapunov exponent of input pricing (Type I market)

β	0.0001	0.0005	0.00005	0.001	0.005	0.01
mLE	0.0128	0.0160	0.0168	0.00391	0.00205	0.0

5.2 Type II market

Then, the same procedure to check the chaoticity is applied to the pricing generated by the Type II market. In the market, we assume the function $W(\lambda)$ including the capacity μ is described by the ordinary mean waiting time used for the characterization of queueing system.

Figure 6 shows an example of pricing time series of artificial market. .

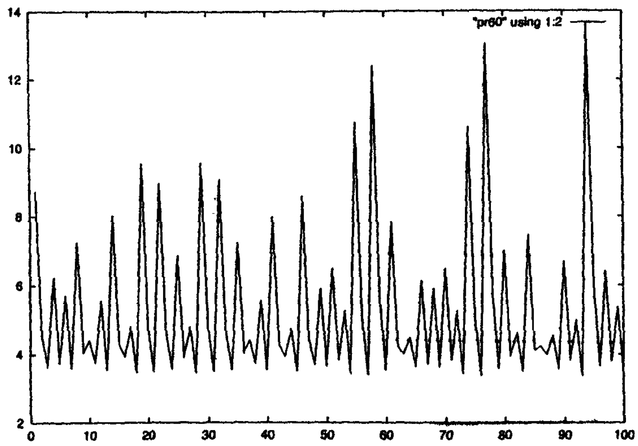

Fig.6-Example of pricing time series of artificial market

We use M/M/1 and M/E(k)/1 models for $W(\lambda)$. Table 2 summarizes the mLE for several μ. We see from table 2, if the capacity is $0.6 < \mu < 0.9$,then the pricing time series becomes chaotic.

Table 2.Maximum Lyapunov Exponent of the input pricing

μ	0.4	0.5	0.6	0.7	0.8	0.9	1.0
M/M/1	-0.0502	-0.000431	0.0293	0.0161	0.0216	0.00295	-0.00263
M/E(k)/1	-0.0130	0.0109	0.0426	0.0445	0.0101	0.0124	0.00685

6 Fractality of pricing (Type I market)

6.1 Conditions for pricing to be fractal

In the following, we examine the conditions of artificial market for the pricing to be fractal focusing on the role of type 5 agents and LCS scheme, only for the Type I market. More precisely, if the numbers of type 5 agents increase the artificial pricing to be resemble to real stock price. On the same way, if type 1,2,3,4 agents employ LCS scheme in the decision making, the artificial pricing can emulate real pricing such as stock prices.

It is known that very typical feature of real pricing is the time series is fractal. For example, usually, the fractal dimension is of real stock price is close to 1.5 (the fractal dimension observed in Brownian motions), and ideal stock price is modeled by the Brownian motion.

For testing the time series to be fractal, we utilize the logarithm of the wavelet transform coefficients of pricing by depicting the value along the axis of dilation index.

$$x_n^m = \int x(t)\Phi_n^m(t)dt \tag{11}$$

$\Phi_n^m(t)$ is defined by using the scale and shift transform of the basic function $\Phi(t)$ as follows. The numbers n, m mean the indices of the scale and shift transform (dilation and transform) , respectively.

$$\Phi_n^m(t) = 2^{m/2}\Phi(2^m t - n) \tag{12}$$

Then, we have the relation as follows (Tokinaga.et al.l 1996, Wornell 1993) by using fractal dimension D .

$$var(x_n^m) = \sigma^2 2^{-\gamma m}, \gamma = 5 - 2D \tag{13}$$

Therefore, if the variance of x_n^m is calculated, then the variance and the fractal dimension of the time series can be estimated. By taking the logarithm of equation (13), we have a linear function of index m. Then we adopt a linear regression curve to the logarithm of variance, and define the root mean square error of the difference as

$$R_w = \sum[(\log(var(x_n^m) - c_0 - c_1 m)^2]^{1/2}/(MX_r) \tag{14}$$

where M is the range of index m, and X_r is the difference between the maximum and minimum of $\log(var(x_n^m))$. If the time series $x(t)$ is fractal, then $R_w = 0$ theoretically.

6.2 Role of LCS

So far, we assumed that the agents do not use the LCS scheme in the decision making. Namely, they use the individuals with highest fitness selected from the pool. The assumption is necessitated to explain the chaoticity.

However, for the test of fractality, the randomness in agents' behavior is expected to give good effect on the reality of artificial stock price. We employ a method for the explanation of LCS effect on the stock price by changing the range of selection of individuals from the pool. We define pS as a probability in selecting an individual from the pool.

In the original LCS scheme, by calculating the fitness of individual, we select N_G individuals having relatively higher fitness. We assume agents of type 1,2,3,4 use one of the individual at random from N_S individuals which are selected from the top of N_G members in the pool. The probability p_S is therefore defined as $p_S = N_S/N_G$. If $p_S = 1$ the method for the usage of individual is the original LCS scheme. If $N_S = 1, p_S = 1/N_G$, then the method corresponds to the exclusion of LCS scheme, in which agents use only individuals with highest fitness.

The result for several p_S is summarized in Table 3 by showing the value of R_w for testing fractality. As is seen from Table 3, if we choose $p_S > 0.7$, the artificial stock price is said to be fractal.

Table 3-Relation between fracatility and p_S

p_S	1.0	0.8	0.6	0.4	0.2	No LCS
R_w	0.00649	0.00781	0.0914	0.01392	0.0243	0.0138
D	1.613	1.765	1.782	1.747	1.726	1.730

7 Conclusion

We consider the dynamic behavior of agent-based artificial market by using the Genetic Programming and its applications. Five types of agents are introduced and within them a number of agents are assumed to be heterogeneous and to have their own rule for predicting the future pricing. We showed that the pricing time series bears chaoticity under certain condition, however, becomes to be fractal by changing the ratio of the number of agents in each type.

The problems remained to be solved include the extension of the method to various real time series and the improvement of control. Further works will be continuously done by the authors.

References

[1] Chen.X. and Tokinaga.S. (2002),Approximation of chaotic dynamics for input pricing at service facilities based on the GP and the control of chaos , Trans. IEICE, vol.E85-A,no.9,pp.2107-2117.

[2] Chen.X. and S.Tokinaga.S.(2003), Synthesis of multi-agent systems based on the co-evolutionary genetic Programming and its applications to the analysis of artificial markets, Trans. IEICE,vol.E86-A,no.10,pp.1038-1048.

[3] Chen.X. and S.Tokinaga.S. (2004),Multi-agent modeling of artificial stock markets by using the co-evolutionary GP approach, Journal of Operations Research Society of Japan, vol.47, no.3, pp.163-181.

[4] Holland.J.H. and Miller.J.H.(1992),Artificial adaptive agents in economic theory, Learning and Adaptive Economic Behavior, vol.81,no.2, pp.365-370.

[5] Ikeda.Y. (2002),Estimation of the chaotic ordinary differential equations by co-evolutionary genetic programming, Trans. IEICE, vol.E85-A,no.4,pp.424-433.

[6] Ikeda.Y. and Tokinaga,S.(2000),Approximation of chaotic dynamics by using smaller number of data based upon the genetic programming, Trans. IEICE, E83-A,no.8,pp.1599-1607.

[7] Ikeda.Y. and Tokinaga.S.(2001),Controlling the chaotic dynamics by using approximated system equations obtained by the genetic programming, Trans. IEICE,vol.E84-A, no.9,pp.2118-2127.

[8] Ikeda.Y.(2002),Estimation of the chaotic ordinary differential equations by co-evolutionary genetic programming", Trans. IEICE, vol.E85-A,no.4, pp.424-433.

[9] Ikeda.Y. and Tokinaga.S.(2003),Synthesis of multi-agent systems based on the co-evolutionary genetic Programming by considering social learning and its applications to the analysis of artificial stock markets, Journal of Japan Society for Management Information, vol.12,no.3,pp.17-35.

[10] Ikeda.Y. and Tokinaga.S.(2001),Controlling the chaotic dynamics by using approximated system equations obtained by the genetic programming, Trans. IEICE,vol.E84-A, no.9,pp.2118-2127.

[11] Ikeda.Y. and Tokinaga.S.(2004), Chaoticity and fractality analysis of an artificial stock market by the multi-agent systems based on the co-evolutionary Genetic Programming, Trans. IEICE,vol.E87-A, no.9,pp.2387-2394.

[12] Koza.J.R.(1990),Genetic programming:A paradigm for genetically breeding populations of computer programs to solve problems, Report No.STAN-CS-90-1314, Dept.of.Computer Science Stanford University.

[13] Koza.J.R.(1991),Evalution and subsumption using genetic programming, Proc of the First European Conference on Artificial Life, MIT Press.

[14] Koza.J.R.(1992),Genetic Programming,MIT Press.

[15] LeBaron.B.(2002),Agent-based computational finance: suggested readings and early research, Journal of Economic Dynamic and Control,vol.24,pp.679-702.

[16] Li.T.Y. and Yorke.J.A.(1975),Period three implies chaos", Am.Math.Mon.,vol.82,pp.985-992.

[17] Rump.C.M. and Stidham,S.Jr,(1998),Stability and chaos in input pricing for a service facility with adaptive customer response to congestion,Management Science,vol.44,no.2,pp.246-261.

[18] Stidham,S.Jr(1992),Pricing and capacity decisions for a service facility:Stability and multiple local optima",Management Science,vol.38,no.8,pp.1121-1139.

[19] Tesfatsion.L.(2000),Introduction to the special issue on agent-based computational economic, Journal of Economic Dynamic and Control,vol.25,pp.281-293.

[20] Tokinaga.S.et al. (1996),A forecasting method for time series bearing fractal geometry and its application,Trans. IEICE,vol.J79-A,no.11,pp.1793-1800 .

[21] WornellG.W.(1993),Wavelet-based representation for the 1/f family of fractal process", Proc. IEEE, vol.81,no.10,1428-1450..

(Professor, Faculty of Economics Kyushu University)