

## Association for Information Systems AIS Electronic Library (AISeL)

---

PACIS 2001 Proceedings

Pacific Asia Conference on Information Systems  
(PACIS)

---

December 2001

# An Agent-Based Supply Chain Management Model

Ken Taniguchi  
*University of Tsukuba*

Setsuya Kurahashi  
*University of Tsukuba*

Takao Terano  
*University of Tsukuba*

Follow this and additional works at: <http://aisel.aisnet.org/pacis2001>

---

### Recommended Citation

Taniguchi, Ken; Kurahashi, Setsuya; and Terano, Takao, "An Agent-Based Supply Chain Management Model" (2001). *PACIS 2001 Proceedings*. 6.  
<http://aisel.aisnet.org/pacis2001/6>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2001 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# An Agent-Based Supply Chain Management Model

Ken Taniguchi, Setsuya Kurahashi, Takao Terano

Graduate School of Systems Management, University of Tsukuba, Tokyo  
3-29-1, Otsuka, Bunkyo-ku, Tokyo 112-0012, Japan  
{taniguti, kurahasi, terano}@gssm.otsuka.tsukuba.ac.jp

## Abstract

This paper proposes agent-based formulation of a Supply Chain Management (SCM) system for manufacturing firms. We model each firm as an intelligent agent, which communicates each other through the blackboard architecture in distributed artificial intelligence. To overcome the issues of conventional SCM systems, we employ the concept of information entropy, which represents the complexity of the purchase, sales, and inventory activities of each firm. Based on the idea, we implement an agent-based simulator to learn 'good' decisions via genetic programming in a logic programming environment. From intensive experiments, our simulator has shown good performance against the dynamic environmental changes.

**Keywords :** Supply Chain Model, Genetic Programming, Agent-Based Simulation, Information Complexity

## 1. Introduction

The supply chain of a manufacturing enterprise is a world-wide network of suppliers, factories, warehouses, distribution centers and retailers through which raw materials are acquired, transformed and delivered to customers. In order to optimize performance, supply chain functions must operate in a coordinated manner. But the dynamics of the enterprise and the market make this difficult: materials do not arrive on time, production facilities fail, customers change or cancel orders, etc. causing deviations from plan. In these situations, Supply Chain Management (SCM) (Curran 1999, Handfield 1998) has become increasingly important, as it is seen as one of the key determinants in achieving competitive advantage for manufacturing enterprise.

Nowadays, what required is the autonomous decentralized SCM, which allows each site to aim at its optimum plan autonomously and at the same time to consider the optimum as a whole. When production and distribution lines are frequently modified like today, it is impractical to centrally manage or plan the characteristics and the constraints of each site whenever a modification is made. The huge amount of cost must be spent only for the software maintenance, and this is one of the reason that only small portion of companies have actually implemented SCM. The autonomous decentralized SCM must be an agent-oriented system with organizational intelligence, which allows each agent to consider its own characteristics and at the same time to concern about the optimum as a whole.

This paper proposes agent-based formulation of a SCM system for manufacturing firms. Our model was inspired by the studies of (Efstathiou, et al. 1999) and (Sivadasan, et al.

1999). They have observed that the performance of SCM systems can be measured by the concepts of the information entropy. We have also carried out the research on agent-based simulation for social and organizational problems (Epstein 1996, Takadama, et al. 2000) using Genetic Algorithms (Mitchell 1996). Based on this, we model each firm as an intelligent agent, which communicates each other through the blackboard architecture in distributed artificial intelligence.

The objectives of the paper are twofold. First, using agent-based simulation techniques with the information entropy method, we will model various aspects of decentralized SCM systems. Second, using GA-based techniques, especially Genetic Programming with logic programming environments, we will show autonomous control mechanisms among the firms would emerge. The rest of the paper is organized as follows: In Section 2, we explain the background and motivation of the research. In Section 3, we discuss why and how we model autonomous decentralized SCMs. In Section 4, we briefly survey the agent-based technology applied to SCM modeling. In Section 5, we will show the first simulation results to explain how the information entropy model works well. In Sections 6 and 7, we will demonstrate the second simulation results, in which we employ GP techniques. Finally, in Section 8, some concluding remarks will be given.

## **2. Background**

The manufacturing industry is in rush to rebuild supply chains. The various production management methods are adopted today such as: JIT (Just In Time), which "supplies the right things, at the right time, at the right amount, to the right place", MRP (Material Requirement Planning) in bucket style, MRP II (Manufacturing Resource Planning) as an expanded version of MRP, or DRP (Distributed Resource Planning) which applies the idea of MRP II to distribution processes. However, the manufacturing industry is facing the problems that can not be addressed with these production management methods. The tastes of consumers change every minute, and what required now is the supply chains, which enable us to follow the changes of the market demand. This is shown as a shift to the market-in type of business models, as typified by the direct production model of PC manufacturers.

In global supply chains, the synchronization between sales, production, and supply can be globally implemented only if there is a network, which connects to the worldwide business sites and collects the information in real-time basis. Each production site should correspond with its marginal lead-time to minimize the stock, deliver the products on time, and at contingency, the second optimum supply plan need to be rebuilt based on the constraints of every processes. The production plan may be disarranged due to internal and/or external disturbances, such as revision of purchase order, incorrect demand forecast, late delivery of parts, or defective products.

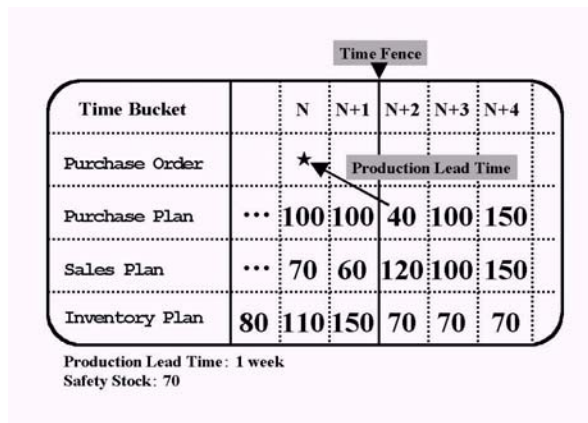
In these situations, supply chain systems need to be managed optimally. The current SCM (Supply Chain Management), however, adopts the centralized style of architecture based on a high speed network, which connects the sites distributed worldwide. The real-time information from the sites, such as goods in process or sales, need to be collected into the database located at the headquarter, then the optimal production plan or the demand forecast can finally be worked out. However, one of the problems that manufacturers are facing is the relocation of labor-intensive production lines to developing countries. Factories are getting

out from the information-driven society. The production systems are subtly differ between the sites, and each site is promoting BPR (Business Process Re-engineering) desperately but in its own way. This is one of the characteristics of Japanese production system.

The SCM needed for such autonomous decentralized production systems is not controlled in the conventional centralized manner. What required is the autonomous decentralized SCM, which allows each site to aim at its optimum plan autonomously and at the same time to consider the optimum as a whole. When production and distribution lines are frequently modified like today, it is impractical to centrally manage or plan the characteristics and the constraints of each site whenever a modification is made. The huge amount of cost must be spent only for the software maintenance, and this is one of the reason that only small portion of companies have actually implemented SCM. The autonomous decentralized SCM is an agent-oriented system with organizational intelligence, which allows each agent to consider its own characteristics and at the same time to concern about the optimum as a whole.

### 3. Description of Autonomous Decentralized Supply Chain Model

The model proposed on this paper is different from the conventional centralized SCM. In this model, agents are assigned to distributed sites and autonomously adjust the purchase plan. The basic purchase plan is based on MRP(Material Requirements Planning) or DRP(Distribution Resource Planning) in order to achieve more realistic SCM. The purchase plan, as shown in Figure 1, is determined by purchase (or production), sales (or demand), and safety stock, which are calculated for each time bucket. The demand represents the intermediate goods required for the production in subsequent processes. The quantity of purchase is determined based on the predefined safety stock for each process.

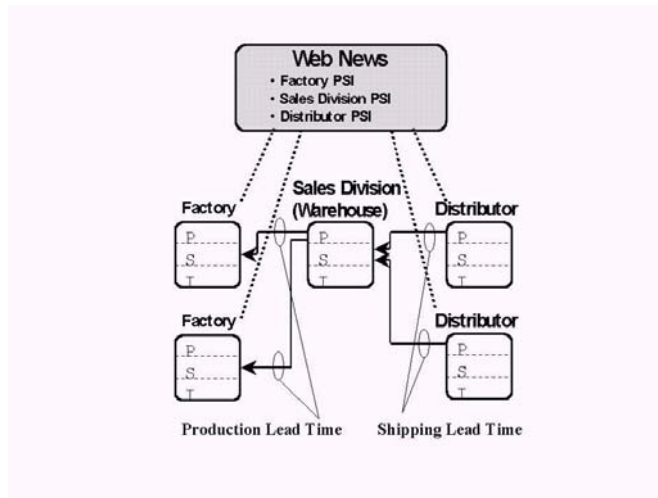


**Figure1: Example of PSI (Purchase, Sales, and Inventory) Table**

The modification is allowed only for the buckets after the time fence. The configuration of the entire supply chain is shown in Figure 2. In the model like this, where MRP and DRP are combined, warehouses and sales sites can be treated in the same way to treat production processes. As indicated as the connections between processes, the Sales (or Shipment) 'S' of the previous process is corresponding to the Purchase (or Production) 'P' of the subsequent process.

The Web News located at the center of Figure 2 contains the 'PSI' (Purchase, Sales, Inventory) tables of the processes (abbreviated as 'PSI' below). Each process looks up this Web News to keep track of the situation of the other processes and to adjust the own

purchase plan based on this information. For example, when the production in a factory has been delayed, it is reflected to its result on the Web News and the other processes can recognize it.



**Figure 2: Autonomous Decentralized Supply Chain Model**

In centralized SCM, the adjustment of entire production plan has been made at the position where corresponds to this Web News. This model leaves it up to each process. So the adjustment shall be autonomously made only by the processes, which requires the adjustment in response to the changes on other processes or the changes on connections of processes.

The degrees of differences between plan and result are considered as 'complexity' of supply chains (e.g., Sivadasan et al. 1999). There is no need of SCM if demand and purchase result are exactly as forecasted and planed. The real-time adjustment on purchase plan is required as a result of various external disturbances, such as demand fluctuation or late delivery of materials, and internal disturbances, such as delay on production or defective products. The purpose of SCM is to minimize the differences between the planed values and the resulted values.

In our model, such complexity is considered as information entropy. The following formula is provided to calculate it as the quantity of information.

$$H = - \sum_{i=1}^n p_i \log_2 p_i$$

Here, the entropy,  $H$ , is defined as the uncertainty and variability of the system associated with a set of  $n$  events for which  $p_i$  is the probability of  $i$ -th event occurring. Instead of being satisfied with the purchase plan provided by MRP or DRP, each process tries to minimize the difference between plan and result by adjusting own production plan based on the information on the Web News.

When you depend only on MRP or DRP, the upstream processes are planed based on PSI plan of several periods ahead, which means that the upper processes are depending on more uncertain demand forecast than the lower processes (Hieber, et al. 1999). This directs our attention to try to improve the accuracy of demand forecast, but it is extremely difficult to forecast the accurate demand in the real market. Then our focus is shifted on how the

production system(or the purchase order system) can flexibly correspond to the uncertain demand, and the production(purchase) plan, which anticipates the demand fluctuation, is required. Our model is trying to reproduce the professional work of 'process artisan', who handles subtle production adjustments like an act of God.

#### **4. Agent Technology in SCM Applications**

One of the studies that apply intelligent agents to SCM is the model, which uses auction between agents as a method for resource allocation (Mori et al., 1998). In order to follow promptly to the changes of production environment that is interlocked with the market trend, the resource allocation must be well coordinated between each department within the supply chain. In this model, this problem is addressed using intelligent agents. There are four agents: Management Agent, Account Agent, Resource Agent, and Production Agent. Their respective functions are: to compute the resource allocation using linear programming and manage the entire supply chain, to manage products and customers and conduct auctions, to forecast demand and place orders for materials, and to manage the capacity of each production process. The allocation of resources is readjusted through the auctions between Account Agents in order to maximize their internal profit for the future. Since this model put an emphasis on the resource allocation, synchronization of production processes and reduction of stocks in process are not considered. Also, the model does not consider the difference between production plan and actual orders, or between planned production and resulted production reduced by disarrangement of production lines. With this model, it should be difficult to address the actual problems in shop-floor, such as the increase of goods on hand or goods in process, which may occur even if the resources are well allocated.

The studies, which applied multiagent model to assembly lines type of supply chains (Schlueter-Langdon 2000), (Lin, et al., 2000) indicated the value of information using Swarm, an evolutionary simulation tool. It is indicated that the conventional decision models exerted their effect on hierarchical production systems, but the techniques such as social simulations are needed for today's production systems like supply chains, where decentralized instructions and controls are required. They conducted simulations on production lines for assemblies, such as personal computers or consumer electronics, and compared the results between three production methods. It was found from the result that the production method which makes a final assembly in response to order is superior to the other two methods, production for stock and production on orders, in the points of inventory cost and cycle time. They also indicated that the demand information reduces inventory cost (information replaces stock). In this model, however, the ways to handle demand information and manufacture information were not practical. This is because the supply chain network was combined with information such as orders, stocks, or production and physical distribution, but it was not combined with the basic production plan. In the methods like MRP, which are widely adopted in actual production processes, the production is planned based on those demand and manufacture information.

Regarding supply chain network, we referred to following articles (Hieber, et al., 1999), (Sterman 1989). Concerning supply chain complexity, we relied on (Sivadasan, et al., 1999) and (Efstathiou et al, 1999).

#### **5. Agent Simulation Experiments**

This section shows how the information entropy changes when agents suffer the fluctuation of demand. Here we consider the case where the supply chain is comprised from three agents (factory, sales division, and distributor) and the goods manufactured at the factory are received at the distribution warehouse of the sales division, then delivered to the distributor.

### 5.1 Basic Information Reported on Web News

The basic information about agents is reported on Web News in the following format.

- Factory
  - Agent name: a
  - Safety stock: 0
  - Logistics pattern: a --> b
  - PSI information: (omitted)
- Sales Division
  - Agent name: b
  - Safety stock: 100
  - Logistics pattern: b --> c1
  - Purchase lead time: 6 (days)
  - PSI information:
    - Bucket : .., -2, -1, 0, 1, 2,..
    - P plan : ..,100,100,100,100,100,..
    - P results: .., 90,120, 0, 0, 0,..
    - S plan : .., 70, 70, 70, 60, 90,..
    - S results: .., 60,140, 0, 0, 0,..
    - I : ..,100, 80,110,150,160,..
- Distributor
  - Agent name: c1
  - Safety stock: 0
  - Purchase lead time: 1 (day)
  - PSI information: (omitted)

P, S, and I of PSI information stand for Purchase (or Production), Sales (or Shipment), and Inventory respectively. Bucket represents the unit of time for managing PSI, and the unit can be the period like month, week, or day. PSI information is time-series data, which indicates the PSI result/plan value for each bucket. Safety stock is a quantity of safety stock, Purchase lead time is a period takes for Purchase (in this case, in days), and Logistics pattern is direction of distribution.

### 5.2 Defining Disturbance at each firm

The information entropy is the value to be an indicator when the degree of demand fluctuation described in PSI information is considered as complexity of supply chains. The process of calculation is described below.

1. Calculates fluctuation difference by bucket
  - Calculates the differences between resulted values and planed values by bucket, and the differences between values of previous plan and latest plan by bucket, using the values in P of PSI information.
2. Calculates occurrence probability by grid

Allocates the computed fluctuation differences to each grid and then calculates the probability that the fluctuation difference occurs within the targeted range for each grid based on the frequency of occurrence. The grid means the range of fluctuation difference (when the width of the grid is 10, for example, then the ranges are 0-9, 10-19, 20-29,...) and must be determined in advance.

3. Calculate occurrence probability by bucket  
Calculates the probability that the fluctuation difference occurs in each bucket based on the fluctuation difference by bucket and the occurrence probability by grid.
4. Calculate information entropy  
Based on the occurrence probability by bucket, calculates the information entropy using the formula shown in the previous section.

### ***5.3 Measuring Bull-whip Effects***

For example, when the orders 100 come in against the sales plan 70 in bucket 1 (next day) of the distributor (Agent c1), the information propagation between agents will be carried out as described below.

1. In the distributor's *PSI information*, the value of *S plan* in bucket 1 is changed from 70 to 100, then the difference of 30 will be added to the value of *P plan* in the same bucket accordingly.
2. Since the *Purchase lead time* from distributor to sales division is 1 (1 day), the sales division's *S plan* in the bucket 0 (sales plan of today) will be changed from 70 to 100. The *Purchase lead time* from sales division to factory is 6 (6 days), which means that it is not allowed to change the sales division's *P plan* up to the bucket 5. Therefore, the difference 30 of the sales plan will be reflected to the bucket 6 (which is equivalent to time fence). The sales division's *P plan* in the time fence bucket is determined based on the values in *S plan* and *Safety stock*.
3. The value of the sales division's *P plan* in the time fence bucket 6 will be reflected to the factory's *S plan* in bucket 0 and then the value of factory's *P plan* will be changed accordingly.

In this manner, the fluctuation of demand information in downstream is propagated toward upstream and its fluctuation differences are reflected to the information entropy of each agent.

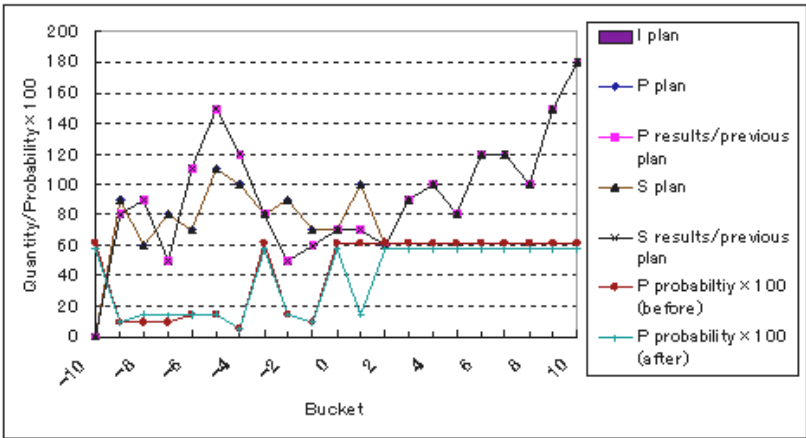
### ***5.4 Validation of the Information Theoretic Method***

We have performed a simulation using given set of *PSI information* (time-series data) of each agent and generating the information propagation. The changes on the *PSI* value and the occurrence probability of each bucket before and after the information propagation are shown in Figures 3, 4, and 5. The grid width was set to 10. From the Figures, we observe the Bull-whip Effects: the demand changes of the distributor (Agent c1) are propagated to the upstream of the chains, the sales division (Agent b) and the Factory (Agent a).

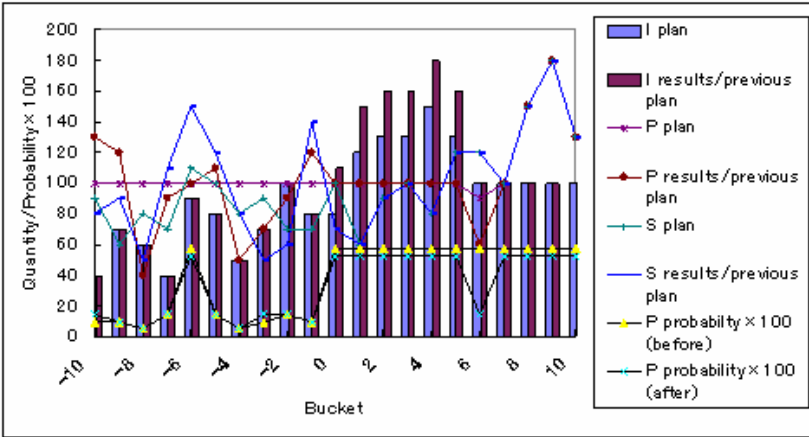
The Effects are directly measured by the change of information entropy. In Fact, Table 1 shows the changes on the information entropy before and after the information propagation. Each agent shows the increase of information entropy after the propagation. The result



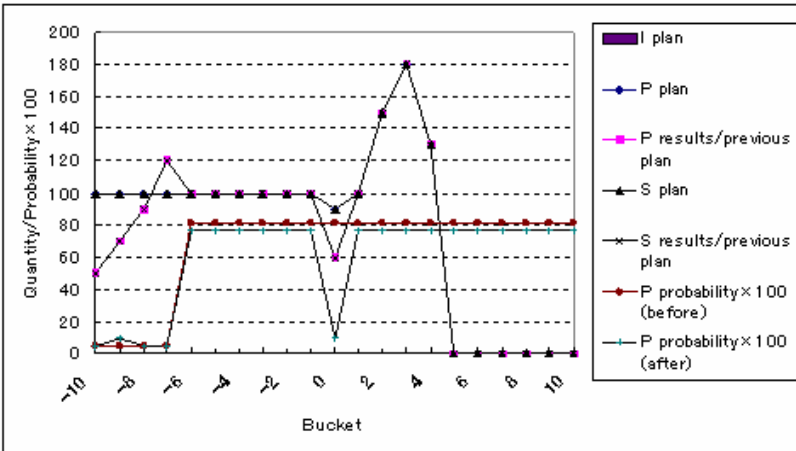
indicates that all of the agents in the supply chain are affected by the fluctuation caused by the order placement from the distributor, and that the complexity of the supply chains itself, or the indeterminacy of fluctuation, is amplified.



**Figure 3: Demand Fluctuation of the Distributor (Agent c1)**



**Figure 4: Demand Fluctuation of the Sales Division (Agent b)**



**Figure 5: Demand Distribution of the Factory (Agent a)**

In this simulation, the safety stock of distributor was set to zero for simplicity. In actuality, however, the orders from distributor to sales division are placed under the consideration of safety stock in most cases. If multiple distributors exist in the situation, the further increase of the information entropy is anticipated.

**Table 1: Information Entropy Before and After the Propagation**

<i>Agent</i>	<i>Before</i>	<i>After</i>
c1(Distributor)	8.27	8.80
b (Sales Div.)	8.44	8.85
a(Factory)	5.03	6.06

## 6. Applying Genetic Programming Method with Logic Programming Techniques

In this section, we propose genetic programming (e.g., Koza 1992) with Horn clauses (e.g., Leung 1995a) as a simulation method to reduce the complexity of SCM. Since the analysis of generated programs is easier, the genetic programming method was employed. Since diversity can be given to solutions by generating redundant solutions through backtrack, the logic programming method was employed. When the search space might be multimodal landscape, a population in GP could immediately converges to proximity of local optimum since conventional GPs do not have mechanisms to maintain multiplicity in tree structure.

Thus, In order to maintain multiplicity in tree structure, we introduce Messy GA style genetic expressions and genetic operations to GP proposed by (Kargupta, 1995). In other words, we bring in the processes of multiple definitions and shortage definitions to GP mechanisms using index of gene locus.

We also use Horn clauses for the expression of tree structures as mentioned above, and the programs generated with GP are also expressed in Horn clauses. In the case of a Horn clause with nodes corresponding to gene loci of multiple definitions, normally, the first appearance of it is given the priority. However, the other solutions, which reflect the nodes with lower priority can be generated by implementing, backtrack using logic programming.

We designed two models for solving targeted problems with cooperation between multiple agents. One is a homogeneous model, where all of the agents operate with the same program, and another is a heterogeneous model, where each agent operates by referring different programs. For example, the following Prolog program is represented in a tree form.

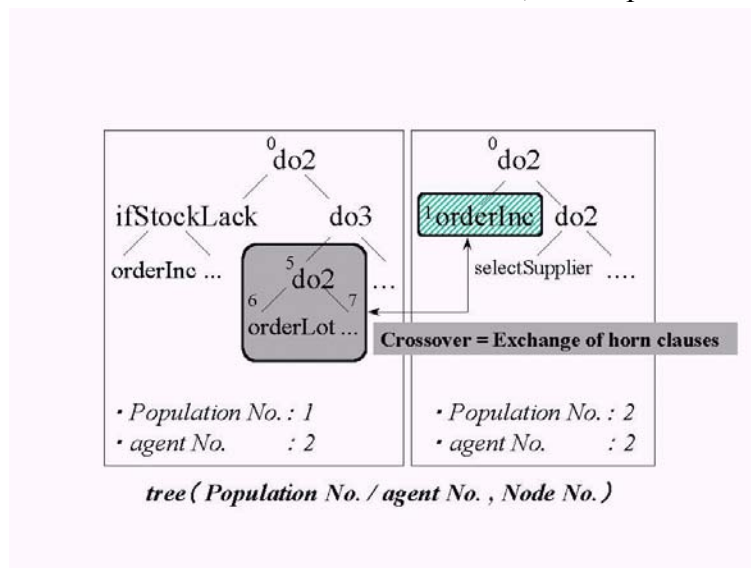
```
function(tree(1/agent1,0),f1) :-
    terminal(tree(1/agent1,1),a),
    function(tree(1/agent1,2),f2).
function(tree(1/agent1,2),f2) :-
    terminal(tree(1/agent1,3),b),
    terminal(tree(1/agent1,4),c).
terminal(tree(1/agent1,1),a).
terminal(tree(1/agent1,3),b).
terminal(tree(1/agent1,4),c).
```

The predicate `function/2` is used to express a non-terminal node and the second argument represents the node (`f1, f2`). The predicate `terminal/2` is used to express a

terminal node and the second argument again represents the node (a, b, c). The numbers assigned for individuals (trees) in the population, and the node number corresponds to index of gene locus in Messy GA are expressed in the first argument of predicate function/2 or terminal/2 in the format of `tree(Individual Number/Agent Number, Node Number)`. The Agent Number is an identifier that enables to support the heterogeneous model.

On the tree structure expressed with Horn clauses, the genetic operations are conducted in the following ways.

1. Generates the initial population of random trees expressed with Horn clauses, using available function symbols and terminal symbols. In the case of heterogeneous model, one individual implies a set of trees, which represents each agent.
2. Computes the goodness of fit of each random tree in the population.
3. Applies selections, crossovers, and mutations to form a new population. When the crossover is applied, the crossover point (node number) at the parent is selected at random and the partial tree under the point is exchanged in order to generate two children (refer to Figure 6). In the case of heterogeneous model, the Restricted Breeding method, which makes a crossover between the corresponding trees (agents), is applied. When the mutation is applied, a node of a tree is selected at random and the partial tree under the selected node is exchanged with a partial tree generated at random.
4. Checks the exit criteria and exits if it is satisfied. If not, then repeat the steps 2 and 3.



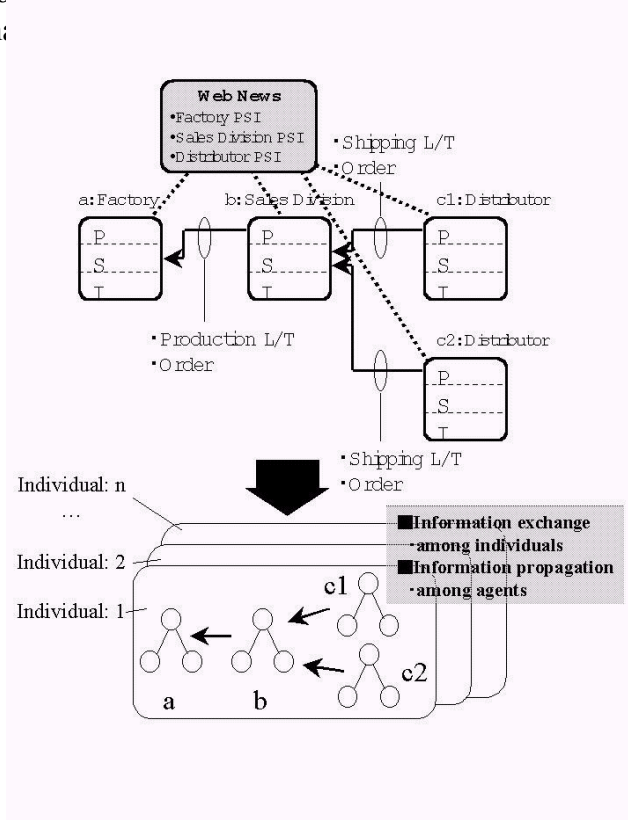
**Figure 6: Crossover Operation Applied to GP Trees**

We try to contrive a model, which controls the aforementioned information entropy as an objective function of entire supply chains. What we are aiming at is to implement the planning of PSI plan values that minimize the information entropy, using this emergent computational method. The agent-based supply chain model using GP Tree Structure is shown in Figure 7.

## 7. Experiment

We have implemented agents or firms with GP functions as described in the previous section. We initially set "Web News" contents with the PSI information of each agent. Then, we

have simulated the model with different configurations of firms when they have placed orders of the firm:



**Figure 7: Agent-Based Supply Chain Model using GP Tree Structures**

The objective of the simulation experiments is to find out an adequate Supply Chain Network or the configuration of the agent communication network in the Supply Chain. The Supply Chain Network should satisfy the optimal PSI relations among the firms measured by the information complexity. Each firm should autonomously determine to which firm and how many orders it makes. We provide each firm with the specific function nodes and terminal nodes for GP, which are described below. The objective function is to minimize the total throughput time.

- **ifStockLack (X, Y)**  
In case present inventory is lower than safety stock, X is executed. If not, Y.
- **do2 (X, Y)**  
Execute X and Y in sequential order.
- **do3 (X, Y, Z)**  
Execute X, Y, and Z in sequential order.
- **orderInc**  
Place the same amount orders a firm receives.
- **orderLot**  
Make the same amount placed orders a firm receives measured by lots.
- **selectSupplier**  
Select a supplier.

Table 2 shows the GP parameters used for the simulation.

Some of the output GP programs are shown in Figure 8. In the Figure, for example, the

upper part of Figure 8 shows the program which represents how a sales firm *c1* behaves. The lower part of Figure 8 shows a production firm *b*. The orders which *c1* has made to *b* is determined by the following equation:

**Table 2: GP Parameters Used in the Simulation**

<i>Parameters</i>	<i>Values or Methods Applied</i>
Population Size	20
Number of Agents	3 (a, b, c1)
Maximum Depth of New Individuals	12
Crossover Rate	0.7
Mutation Rate	0.1
Crossover Strategy	Heterogeneous
Generation Model	Minimal Gap Generation (Sato 1996)

$$\text{Number of placed order} = \text{Number of received order} * X + \text{Number lots of placed order} * Y$$

where, *X* and *Y* respectively means the numbers of orderInc and orderLot. The change of the information entropy when the numbers of the agents are equal to 2, 4, and 8 is also shown in Figures 9, 10, and 11.

```

● Distributor: c1 [Entropy = 8.80]
  ifStockLack((do3((selectSupplier, do3((orderInc, orderInc, orderLot), orderInc), orderLot))

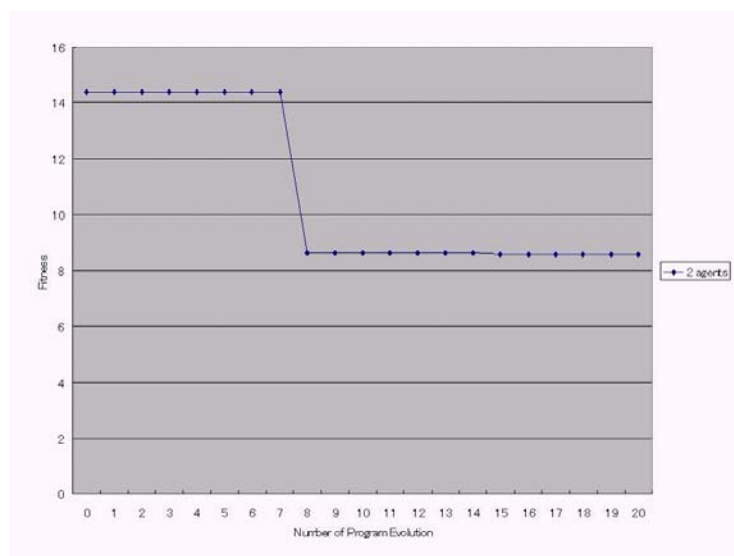
[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10], %Time bucket
[ 0, 90, 60, 80, 70,110,100, 80, 90, 70, 70,100, 60, 90,100, 80,120,120,100,150,180], %Purchase plan
[ 0, 80, 90, 50,110,150,120, 80, 50, 60, 70, 70, 60, 90,100, 80,120,120,100,150,180], %Purchase results/previous plan
[ 0, 90, 60, 80, 70,110,100, 80, 90, 70, 70,100, 60, 90,100, 80,120,120,100,150,180], %Sales plan
[ 0, 80, 90, 50,110,150,120, 80, 50, 60, 70, 70, 60, 90,100, 80,120,120,100,150,180], %Sales results/previous plan
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]] %Inventory results/plan

● Sales division: b [Entropy= 8.73]
  do3((do2((orderInc, orderLot), do3((do3((selectSupplier, orderLot, orderInc),
  selectSupplier, do3((do3((orderInc, orderLot, orderLot), ..., ...))), ...)))

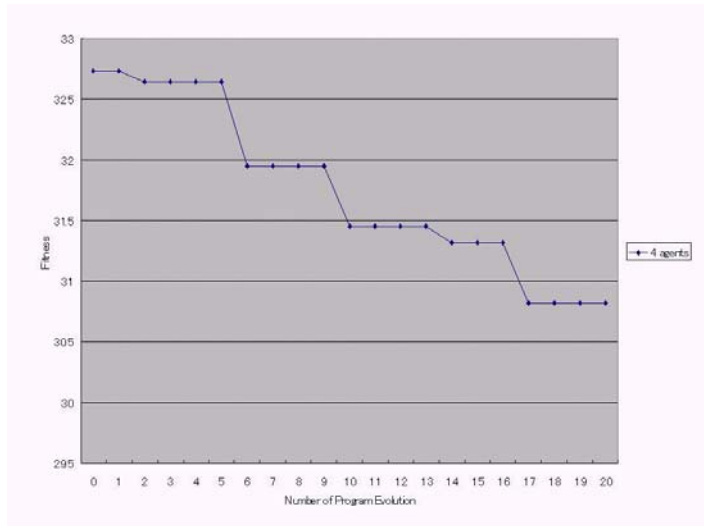
[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10], %Time bucket
[100,100,100,100,100,100,100,100,100,100,100,100,100,100,100, 90,100,150,180,130], %Purchase plan
[130,120, 40, 90,100,110, 50, 70, 90,120,100,100,100,100,100,100, 60,100,150,180,130], %Purchase results/previous plan
[ 90, 60, 80, 70,110,100, 80, 90, 70, 70,100, 60, 90,100, 80,120,120,100,150,180,130], %Sales plan
[ 80, 90, 50,110,150,120, 80, 50, 60,140, 70, 60, 90,100, 80,120,120,100,150,180,130], %Sales results/previous plan
[ 40, 70, 60, 40, 90, 80, 50, 70,100, 80, 80,120,130,130,150,130,100,100,100,100,100]] %Inventory results/plan

```

**Figure 8: Sample Output of the GP Programs**



**Figure 9: Experimental Result I (2 Agents Case)**



**Figure 10: Experimental Result II (4 Agents Case)**



**Figure 11: Experimental Result III (8 Agents Case)**

The sample output of the GP programs is hardly understandable for humans, however, as often the cases in GP literature (Koza 1992), the characteristics are evaluated from the

execution performance: how good they work or how fast they work. From this viewpoint, the results shown in Figures 9, 10, and 11 with 2, 4, and 8 agents cases have revealed that the time for total throughputs are gradually decreasing. This means that they gradually make better decisions about the individual controls of the firms as the programs evolve through the GP iterations. From the experiments, thus, we make the implication that if the agents or members of a SCM system are able to refer to the others' PSI information, they will be able to automatically improve their performances by learning and adaptation mechanisms without any central controls. This is a very desirable suggestion for decentralized autonomous SCM systems.

## 8. Concluding Remarks

In this paper, we discussed that the autonomous decentralized agent method is needed for global supply chains, and demonstrated the effectiveness of complexity index with entropy through the simulation experiments, where the processes such as distributor, sales division, and factory are represented by autonomous software agents. Then we have shown the applicability to improve the efficiency of entire supply chains by reducing this complexity with an emergent computational method, and proposed a novel method of genetic programming to accomplish it.

Future research includes (1) exploring further applications of the proposed agent modeling techniques to the other SCM problems, and (2) extending the architecture to examine the effects of the variety of agents to the result of the simulation.

## References

- T.A.Curran. *SAP R/3 Business Blueprint: Understanding Enterprise Supply Chain Management*, Prentice Hall, 1999.
- J.Efstathiou, F.Tassano, S.Sivadasan, R.Shirazi, J.Alves, G.Frizelle, A.Calinescu. Information Complexity as a Driver of Emergent Phenomena in the Business Community, *Proc. International Workshop on Emergent Synthesis - IWES'99*, pp.1-6, 1999.
- J.M. Epstein, and R. Axtell, R. *Growing Artificial Societies: Social Science from the Bottom Up*, MIT Press, 1996.
- R.B.Handfield. *Introduction to Supply Chain Management*, Prentice Hall, 1998.
- R.Hieber, D.Brutsch, F.Frigo-Mosca. How to Manage your Supply Network to get better results, *Strategic Management of the Manufacturing Value Chain*, pp.289-295, 1999.
- H.Kargupta. *SEARCH, Polynomial Complexity, And The Fast Messy Genetic Algorithm*, Illinois Genetic Algorithms Laboratory report no. 95008, 188pp.,1995.
- J.R.Koza. *Genetic Programming, On the Programming of Computers by means of Natural Selection*, MIT Press,1992.

- F.Lin, T.J.Strader, M.J.Shaw. Using Swarm for Simulating the Order Fulfillment Process in Divergent Assembly Supply Chains, *Economic Simulations in Swarm: Agent-Based Modelling and Object Oriented Programming*, Kluwer Academic Publishers, 2000.
- M. Mitchell. *An Introduction to Genetic Algorithms*, MIT Press, 1996.
- M.Mori, H.Matsuo and M.Kosaka. Resource Allocation Using Intelligent Agent Technology in Supply Chain Management (in Japanese), *Transactions of the Society of Instrument and Control Engineers*, Vol.34, No.11, pp.1675-1683, 1998.
- H.Satoh, M.Yamamura and S.Kobayashi. Minimal Generation Gap Model for GAs Considering Both Exploration and Exploitation, *Proceedings of the IIZUKA'96*, pp. 494-497, 1996.
- C.Schlueter-Langdon. Online Supply Chain Modeling and Simulation, *Economic Simulations in Swarm: Agent-Based Modelling and Object Oriented Programming*, Kluwer Academic Publishers, 2000.
- S.Sivadasan, J.Efstathiou, R.Shirazi, J.Alves, G.Frizelle, A.Calinescu. Information Complexity as a Determining in the Evolution of Supply Chains, *International Workshop on Emergent Synthesis - IWES'99*, pp.237-242, 1999.
- S.Sterman. Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment, *Management Science*, Vol. 35, No. 3, pp.321-339, 1989.
- K. Takadama, T. Terano, K. Shimohara, K. Hori, and N. Nakasuka. Making Organizational Learning Operational: Implication from Learning Classifier System, *J. Computational and Mathematical Organization Theory*, (5:3), 1999-c, pp. 229-252.
- M.L.Wong and K.S.Leung: Combining Genetic Programming and Inductive Logic Programming using Logic Grammars, *Proceedings of the 1995 IEEE International Conference on Evolutionary Computing*, Vol. 2, pp. 733-736, 1995.