

# Optimal Control Strategy for Serial Supply Chain

Min Huang<sup>1,2</sup>, W.H.IP<sup>3</sup>, Xingwei Wang<sup>1</sup> and Jianqing Ding<sup>1</sup>

<sup>1</sup>College of Information Science and Engineering,  
Northeastern University, Shenyang, Liaoning

<sup>2</sup>Key Laboratory of Integrated Automation of Process Industry,  
Northeastern University, Shenyang, Liaoning,

<sup>3</sup>Department of Industrial and Systems Engineering,  
The Hong Kong Polytechnic University, Hung Hom, Kowloon,

<sup>1,2</sup>P.R.China

<sup>3</sup>Hong Kong

## 1. Introduction

With the emerging of global economy and the development of the technology in computer and communications, the enterprises are facing to new opportunities but also more challenges, which led to the concept of SC (Supply Chain). By controlling and collaborating each part of the supply chain, SCM (Supply Chain Management) reaches the aim of reducing the cost, improving the quality as well as service level, and so on, further enhances the integrated competition ability of the whole supply chain. SCM includes many aspects of the management activity, and the inventory management is one of the key aspects among them.

In this chapter, the research focuses on inventory control strategy optimization related to activities, such as the purchase, production, storage and transportation of material, work in process and finished goods inventory within up-stream and down-stream enterprises along the serial supply chain. The main research aspects are as follows:

1. To review the state of art of inventory control strategies of supply chain.
2. Supply chain inventory control strategy optimization is based on the simulation of supply chain inventory system. So, in this chapter, the general model of serial supply chain inventory control strategy is addressed.
3. Model for single objective control strategy optimization is established, which describes the optimization problem of serial supply chain inventory control strategy with the objective of minimum cost and the constraint of the customer service level and average input standard deviation.
4. The algorithm of GA (Genetic Algorithm), Random-PSO (Particle Swarm Optimization) and PEA (Pheromone Evolutionary Algorithm) are designed for the model. Simulation studies suggested that each of the algorithms can solve this problem efficient, and Random -PSO algorithm is most efficient one.

Source: Supply Chain, The Way to Flat Organisation, Book edited by: Yanfang Huo and Fu Jia, ISBN 978-953-7619-35-0, pp. 404, December 2008, I-Tech, Vienna, Austria

## 2. The general control model for serial supply chain

### 2.1 The different control strategies for serial supply chain

Under globalization and the rapid development of computers and information technology, all enterprises face new chances as well as more challenges. This breeds the concept of a serial supply chain, a value-added chain that is composed of a series of enterprises: raw material suppliers, parts suppliers, producers, distributors, retailers, and transportation enterprises. Clients finally get their products, which are manufactured and handled systematically by the enterprises of the chain, started from either the raw material suppliers or the parts suppliers. This series of activities are the total activities of a complete serial supply chain, that is, from the supplier's suppliers to the clients' clients<sup>[1, 2]</sup>.

SCM aims at decreasing the system cost, increasing the product quality, and improving service level by collaborating and controlling the conduct of each entities of the supply chain. The goal is to upgrade the overall competitive ability of the whole system. Hence, the inventory management of the serial supply chain is important. The inventory control strategy of an enterprise affects the cost and the revenue indirectly. Therefore, the target of an optimal inventory is both to maximize the degree of clients' satisfaction and to minimize the overall cost<sup>[3]</sup>.

Inventory decouples the supplying, producing, and selling processes of an enterprise. Each process operates independently. This helps to reduce the effect that comes from the variation of demand forecast, and makes good use of resources when variation happened due to demand changes and market changes. On the other hand, capital is needed for setting up an inventory. The cost includes the capital used for inventory and products in process, the space used for inventory, the expenses on management, maintenance, and discarding of defected products. Inappropriate inventory management even affects the operation efficacy of the enterprise.

The characteristic of uncertainty of a supply chain increases the overall inventory of the whole chain system. It also brings unnecessary cost to the node enterprises of the supply chain. In order to avoid the "bullwhip effect" caused by the uncertainty of demand and supply, the traditional inventory strategy has to be revised. Inventory control of a supply chain can be improved by strategies like shared technology, contract system, and integrated enterprises. Thus, the competitive ability of a supply chain is enhanced.

Generally speaking, there are two kinds of production inventory systems: the push and the pull system. The current worldly popular production inventory control systems of Manufacturing Resource Planning (MRPII) and Just-in-time (JIT) belong to the system, respectively. The push production control system adopts a central control method and organizes production by forecasting the future demand. Therefore, production lead time is estimated in advance. The pull production control system adopts a distributed control method and production is organized according to the real demand<sup>[4]</sup>. Each method has its own advantages<sup>[5, 6]</sup>. Peoples try to combine the two methods to attain better performances<sup>[7-9]</sup>, CONWIP (CONstant Work in Process), proposed by Spearman et al. in 1990, is an example of combined push/pull control method<sup>[10]</sup>. In 2001, Gaury et al. proposed a methodology to customize pull control systems<sup>[11]</sup>. In 2003, Ovalle and Marquez suggested the model of CONWIP control system for a serial supply chain and also shows the corresponding simulation analysis<sup>[12]</sup>. But, up till this moment, all researches on control system of supply chain only deal with single specific control strategy like the push system, the pull system, the classic combined push/pull CONWIP control system<sup>[10, 13-14]</sup>, or the

simple combination of push and pull system. A generally used model of push/pull control strategy and its research is still absent. The aim of the research is to establish a generally used method of the inventory system of a serial supply chain to replace those traditional classic control systems.

In the systems of Kanban and CONWIP, system performances rely on the card quantities. Similarly, in the serial push/pull system, distribution of circulating cards (the number of circulating cards at different stages) determines the control model, guides the production time and production quantity of the generally used system. In the push system, the card numbers between each pair of nodes on the SC is infinite. Therefore, determination of circulating cards becomes a key factor affecting the operating efficacy of a generally used inventory system of a serial supply chain.

This chapter proposes an optimal control model that tackles a series of multi-stages of inventory control system of a supply chain. The model is based on the combination of nonlinear integer programming and the generally used push/pull system of the inventory of a serial supply chain. It determines the distribution of circulating cards by integrating the intelligent algorithms and simulation analysis. Both the case studies have proved that the results from intelligent algorithms are reasonable and effective.

**2.2 The general control model for serial supply chain**

Figure 1 shows that there are  $n$  nodes on the whole serial supply chains. Each node represents upstream/downstream node enterprises like raw material suppliers, manufacturers, distributors, retailers, and clients. Since the final target of a supply chain is to satisfy the clients' demands, each enterprise operates its production and sales under the generally used push/pull inventory system control. That is, production of each node enterprise is affected by the raw material supply of the upstream enterprise and the demand of the downstream enterprise. One important goal of the supply chain is to reach a win-win status, to maximize the profits of the whole chain instead of any individual enterprise. In order to control the quantity of products in process of the chain, there is a fixed product standard on the feedback for the demand on upstream enterprise  $i$  from downstream enterprise  $j$  and marks it as card number  $K_{ij}^u$ . Only when the real quantity of products in

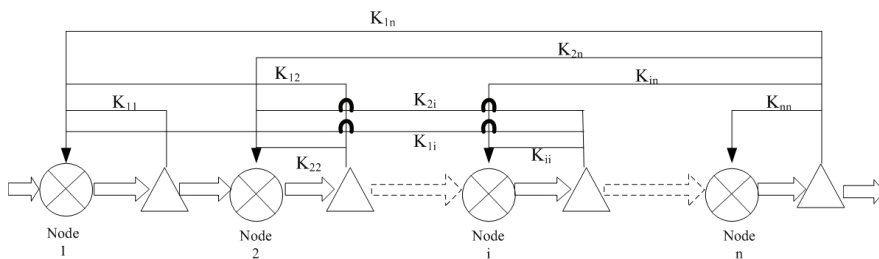


Fig. 1. The general control model for serial supply chain

process of node enterprise  $i$  is less than the forecast product quantity of each downstream enterprise, then it is allowed to proceed with the manufacture. Once the product of a unit is allowed to be manufactured by node enterprise  $i$ , the node's free card is attached to its manufactured container, then the product has finished processing it is sent to the next node enterprise  $i+1$ . The attached circulating card is detached and returns to node enterprise  $i$  as a

free card and authorizes further manufactures of other products. When the value of  $K_{ij}^u$  is  $\infty$ , it means that there is no feedback control on upstream node  $i$  from downstream node  $j$ . If there is no feedback control on node  $i$  from all downstream nodes, then node  $i$  is under the push control.

The commonly used push/pull system has the following properties and assumptions:

1. Clients' demands satisfy the normal distribution with upper and lower bound.
2. The supply chain produces only one kind of product.
3. All upstream enterprises can obtain the return cards from any downstream manufacturing enterprise node without delay (no return delay).
4. There are sufficient materials for the initial node enterprise of the supply chain.

#### Description of the Control Strategy

In order to describe the control strategy, two definitions are needed.

##### Definition 1:

The card number matrix K

$$K = \begin{bmatrix} K_{11} & K_{12} & \dots & K_{1n} \\ & K_{22} & \dots & K_{2n} \\ \cdot & & & \\ \cdot & \dots & \dots & K_{nn} \end{bmatrix} \quad (1)$$

The element  $K_{ij}^u$  of matrix K represents the card number in the control cycle sent by downstream node  $j$  to upstream node  $i$ .  $K_{ij}^u = \infty$  implies that there is no pull control from node  $j$  to node  $i$ . Since all nodes are under the control of their downstream nodes, so the lower triangular matrix of K is meaningless. Thus, values of these elements are fixed as -1. So, we have equation (2):

$$K_{ij} = \begin{cases} -1 & (i > j) \\ 1 \sim K_{ij}^u + 1 & (i \leq j) \end{cases} \quad (2)$$

where  $K_{ij}^u$  represents the upper limit of the card number,  $K_{ij}^u + 1$  represents  $+\infty$ .

##### Definition 2:

Control matrix M: matrix that describes the control model.

$$M = \begin{bmatrix} M_{11} & M_{12} & \dots & M_{1n} \\ M_{21} & M_{22} & \dots & M_{2n} \\ \cdot & & & \\ \cdot & & & \\ M_{n1} & M_{n2} & \dots & M_{nn} \end{bmatrix} \quad (3)$$

Matrix M in (3) shows what kind of control strategy each node of the supply chain has. If the element  $M_{ij}$  of matrix M equals to 1, it means there is pull control on upstream node  $i$  from downstream node  $j$ . If the element  $M_{ij}$  of matrix M equals to 0, it means that there is no pull control on upstream  $i$  from downstream node  $j$ . So we have equation (4):

$$M_{ij} = \begin{cases} 1 & (K_{ij} \neq \infty) \\ 0 & (K_{ij} = \infty) \end{cases} \quad i \leq j \quad (4)$$

**Property 1:** When the sum of all elements of row  $i$  of matrix  $M$  equals to 0, there is no pull control on node  $i$  from all its downstream nodes, which means that node  $i$  is under the push control from its upstream node.

### 2.3 The method for determine the optimal control strategy

The main problem of determining the optimal control strategy is how to choose an appropriate control strategy so that some goals of the supply chain are reached while some constraints are satisfied. A two-level model are presented to cope with this problem. The first-level model is a mathematic programming model, which optimizes the distribution of circulating cards by guaranteeing the goals of the supply chain and satisfying the certain constraints.

The second-level model is the general control model for serial supply chain which is used for the analysis of the inventory system of the whole serial supply chain. Definitions of the variables for analysis is given here and the relationship among the variables is illustrated in figure 2.

#### (1) Logistics variables

- $P_i^t$  quantity of products in process of node  $i$  of period  $t$
- $Y_i^t$  product inventory of node  $i$  of period  $t$
- $S_i^t$  transportation quantity from node  $i$  to  $i+1$  of period  $t$
- $O_i^t$  exported quantity of product of node  $i$  of period  $t$
- $I_i^t$  imported quantity of raw material of node  $i$  of period  $t$
- $X_i^t$  the real usable quantity of raw material of node  $i$  of period  $t$

#### (2) Technology flow variables

- $OP_i^{cust}$  clients' demands of period  $t$
  - $D_i^t$  quantity of inflow orders of node  $i$  of period  $t$
  - $APC_i^t$  number of available cards of node  $i$  of period  $t$
  - $OP_i^t$  quantity of processed orders (manufacturing quantity) of node  $i$  of period  $t$
  - $B_i^t$  quantity of accumulated orders of node  $n$  of period  $t$
  - $DS_i^t$  the expected quantity of transportation of node  $i$  to  $i+1$  of period  $t$
  - $TY_i^t$  quantity of available overall product of node  $i$  of period  $t$
  - $L^i$  period of manufacturing products of node  $i$
  - $Ls^i$  period of transporting products of node  $i$
  - $MLP^i$  product-load ability of node  $i$
  - $UC^i$  vessel capacity of node  $i$
- $K$  matrix of circulating cards, element  $K_{ij}$  represents the number of cards in the control cycle of node  $i$  sent by node  $j$
- $M$  the control matrix mark  $i$ , it is the last node which exerts the pull control on node  $i$

**(3) Capital flow variables**

- $CR_t^i$  quantity of cash demand of node  $i$  of period  $t$
- $ICR_t^i$  inventory value of node  $i$  of period  $t$
- $R_t^i$  compulsory cash-in of node  $i$  of period  $t$
- $Py_t^i$  compulsory payment of node  $i$  of period  $t$
- $Pm_t^i$  product price of transportation unit of node  $i$  of period  $t$
- $Pwip_t^i$  price of products in process unit of node  $i$  of period  $t$
- $mr_t^i$  margin benefits of node  $i$  of period  $t$
- $CumP_t^i$  accumulative benefits of node  $i$  of period  $t$

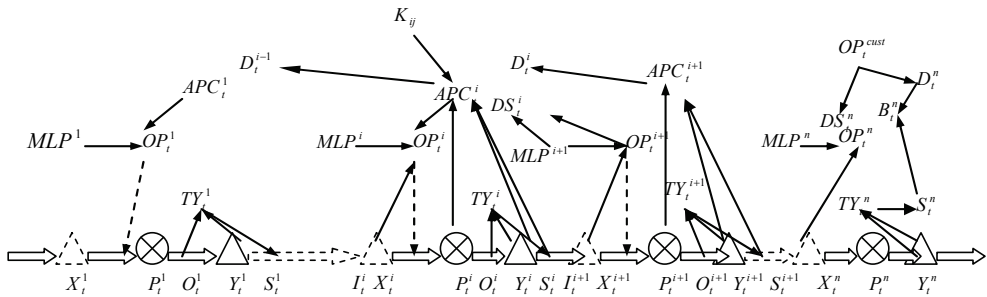


Fig. 2. Illustration of the relationship among the variables

In order to analysis the supply chain, the following performance measurements are defined according to the above relationship among the variables:

1. Service level (%): (the customer satisfy percentage of the supply chain)  
The satisfy percentage of the last node of the supply chain is considered.

$$S_l = \frac{100 * \left( \sum_{t=1}^T D_t^n - B_T^n \right)}{\sum_{t=1}^T D_t^n} \tag{5}$$

2. Standard deviation of inputs:

$$SDO = \sqrt{\frac{1}{T-1} \sum_{t=1}^T \left( OP_t^i - \frac{\sum_{t=1}^T OP_t^i}{T} \right)^2} \tag{6}$$

3. Overall cost of the supply chain:

$$C = \frac{\sum_{t=1}^T \sum_{i=1}^n CR_t^i}{T} \tag{7}$$

The parameters delivered from the first-level model to the second-level model are circulating card distributions of each node. The parameters delivered from the second-level model to the first-level model are the service level, standard deviation of the input, and average overall cost of the supply chain.

### 3. The model and algorithm for optimal control strategy problem

#### 3.1 The model for optimal control strategy problem

Usually, the goal of the supply chain operation is to minimize the total cost with the constraint of service level and input standard deviation. Under this circumstance, the first-level model is a nonlinear integer programming model. It optimizes the distribution of circulating cards by guaranteeing that the average overall cost of the supply chain of the system is minimized, under the constraints of service level  $S_{i_0}$  and the input standard deviation  $SDO_0$ . In another word, solve upper triangular elements  $K_{ij}$  of matrix K. So, the first-level model is given as follows:

$$\min C(K) \quad (8)$$

$$st : \begin{cases} S_i(K) \geq S_{i_0} \\ SDO(K) \leq SDO_0 \end{cases} \quad (9)$$

$$K_{ij} \text{ is the integer between } 1 \text{ and } K_{ij}^u + 1 \text{ where } i = 1, \dots, n, j \geq i \quad (11)$$

where  $K_{ij}$  represents the upper bound of the card number and  $K_{ij} + 1$  represents  $+\infty$ .

The first-level model is a mixture of combination problem and integer programming. If there are  $n$  node enterprises, the card number that needed to be determined will be  $n(n+1)/2$ . When node enterprise  $i$  is under the control of node enterprise  $j$  in the supply chain and the upper bound of card number is  $K_{ij}^u$ , there are  $\prod_{i=1}^n \prod_{j=i}^n (K_{ij}^u + 1)$  states of searching

spaces when the constraints are not considered. To process each state is possible only if both the scale of the supply chain and the cards number are small. Heuristic algorithms are needed to solve practical problems. So intelligent algorithm is applied to solve the first-level model to determine  $k$ . The GA, Random-PSO and PEA are used in this research.

A simulation is used in the second-level according to the general control model for serial supply chain to determine the performance measurements of the supply chain system under a given card distribution  $K$ , because the performance measurements of the supply chain, the cost, the service level and standard deviation of input are implicit functions of card distribution  $k_{ij} (\forall i = 1, \dots, n, j \geq i)$ .

#### 3.2 The intelligent algorithms for optimal control strategy problem

In this section, three intelligent algorithms, GA, Random-PSO and PEA are used designed for the model (8)-(11). The performance of them and the comparison among them are given.

**3.2.1 Genetic algorithm**

This section give the design and analysis of GA for the above model (8)-(11) [15,16].

**3.2.1.1 Coding**

Integer coding is adopted considering the characteristic of the problem. Each bite represents the element of the upper triangular part of matrix K accordingly, and there are  $n(n+1)/2$  bites in total. Figure 3 is the illustration of coding.



Fig. 3. The illustration of coding

The range of each bite is from 1 to  $K_{ij}^u + 1$  where the value of  $K_{ij}^u$  is determined by the ability of each production node. For instance, if the production ability of the first node of a supply chain is 30,  $K_{ij}^u \geq 30$ . If the production ability of the second node is 20,  $K_{ij}^u$  will contain the control cycles of the two nodes, that is,  $k_{12}^u \geq (30 + 20)$ . Let  $K_{ij}^u + 1$  of a certain node as an infinite integer means that its downstream nodes have no limit of card number on it.

Property 2: According to this coding rule, there are different upper and lower bound for each bit of individual as each bit may include different number of nodes.

**3.2.1.2 Fitness Function**

Due to the minimizing property of the objective function, fitness functions. It is obtained by equation (12) as follows:

$$F(K) = f_{\max} - f(K) \tag{12}$$

$$f(K) = C(K) + \alpha_1 * [SDO(K) - SDO_0]^+ + \alpha_2 * [S_{i_0} - S_i(K)]^+ \tag{13}$$

where  $\alpha_1 * [SDO(K) - SDO_0] + \alpha_2 * [S_{i_0} - S_i(K)]$  are the penalties for not satisfying constraints of (9) and (10),  $\alpha_1$  and  $\alpha_2$  are penalty coefficients, and  $f_{\max}$  is a given large value to guarantee that the overall fitness value is non-negative, and

$$[y]^+ = \begin{cases} y & y > 0 \\ 0 & otherwise \end{cases}$$

**3.2.1.3 Operators design**

Each initial solution is obtained by creating an integer within the range of 1 to  $K_{ij}^u + 1$  randomly for each bit.

Two points crossover is adopted here. Two intersecting points are chosen from the chromosomes. Crossover is taken in the space between the two intersecting points and the rest is still inheriting the parent genes.

Mutation is also applied. First, create a number between 0-1 randomly. When the number created is smaller than the mutation probability, mutation will happen by creating an integer that lies within the limits of the circulating cards randomly.

The commonly used roulette mechanism is used as the choice strategy and the biggest iteration number is chosen as the criterion for algorithm termination.



### 3.2.1.4 The Elitist Mechanism

In order to maintain the best chromosome of each generation, an elitist mechanism is used in the choice process. If the best chromosome of the last generation is not duplicated into the next generation, the next generation will randomly delete a chromosome so that the best chromosome of the last generation will be duplicated directly.

### 3.2.1.5 Numerical Analysis

In order to analyze the performance of the algorithm, the problems with 4-nodes ( $10^{18}$ ), 6 nodes ( $10^{40}$ ) are analyzed here..

Example 1 is a supply chain with 4 nodes, the ability of each node enterprise is same, the custom demand is a normal distribution which mean of 4.0 and variance of 1.0, upper and lower bound are 8.0 and 0.0, the parameters are shown in table 1. The supply chain should ensure that the customer service level is not less than 90% and the input standard deviation is less than 2.5.

Node enterprise	1	2	3	4
Quantity of products in process $P_0^i$	8	8	8	8
Product inventory $Y_0^i$	12	12	12	12
Transportation quantity $S_0^i$	4	4	4	4
Exported quantity of product $O_0^i$	8	8	8	8
Imported quantity of raw material $I_0^i$	8	8	8	8
The real usable quantity of raw material $X_0^i$	9	9	9	9
Quantity of processed orders $OP_0^i$	3	3	3	3
Period of manufacturing products $L^i$	1	1	1	1
Period of transporting products $LS^i$	2	2	2	2
Product-load ability $MLP^i$	30	30	30	30
Container capacity $UC^i$	1	1	1	1
Margin benefits $mr_t^i$	1	1	1	1
Product price of transportation unit $Pm_t^i$	2	2	2	2
Price of products in process $Pwip_0^i$	1	1	1	1

Table 1. The initial value of parameters for each node enterprise in 4 nodes problem

According to the parameter set in table 1, the upper-lower bound of card numbers for different control segments are shown in table 2. So the size of this problem is  $41^4 * 71^3 * 106^2 * 141 = 1.602 \times 10^{18}$ , while the constraint is not considered.

The number of nodes	1	2	3	4
Lower and upper bound	1~41	1~71	1~106	1~141

Table 2. The lower and upper bound of card for 4 nodes problem

To analyze the performance of the algorithm, the algorithm is run for 100 times. The best solution is the best one within 100 runs. The best rate is the rate to reach the best value within 100 runs.

Taking reasonable parameter of GA, the best solution  $K=[\infty, \infty, \infty, \infty, 16, 29, 43, 14, 31, 16]$  for 4 nodes problem is obtained.

For the supply chain with 6 nodes enterprises, the parameters are shown in table 3, the upper-lower bound of card number for different problem are shown in table 4.

Node enterprise	1	2	3	4	5	6
Quantity of products in process $P_0^i$	8	8	8	8	8	8
Product inventory $Y_0^i$	12	12	12	12	12	12
Transportation quantity $S_0^i$	4	4	4	4	4	4
Exported quantity of product $O_0^i$	8	8	8	8	8	8
Imported quantity of raw material $I_0^i$	8	8	8	8	8	8
The real usable quantity of raw material $X_0^i$	9	9	9	9	9	9
Quantity of processed orders $OP_0^i$	3	3	3	3	3	3
Period of manufacturing products $L^i$	1	1	1	1	1	1
Period of transporting products $Ls^i$	2	2	2	2	2	2
Product-load ability $MLP^i$	30	30	30	30	30	30
Container capacity $UC^i$	1	1	1	1	1	1
Margin benefits $mr_t^i$	1	1	1	1	1	1
Product price of transportation unit $Pm_t^i$	2	2	2	2	2	2
Price of products in process $Pwip_0^i$	1	1	1	1	1	1

Table 3. The initial value of parameters for each node enterprise in 6 nodes problem

The number of nodes	1	2	3	4	5	6
Lower and upper bound	1~41	1~71	1~106	1~141	1~175	1~210

Table 4. The lower and upper bound of card for 6 nodes problem

The custom demand is a normal distribution with mean of 4.0, variance of 1.0, upper-lower bound of 8.0 and 0.0, constraints condition are service level no less than 90% and the input standard deviation less than 2.5.

The scale of the problem is  $41^6 * 71^5 * 106^4 * 141^3 * 175^2 * 210 = 1.951 \times 10^{40}$  while the constraint is not considered. Taking reasonable population size and iterative number, the best solution  $K=[\infty, \infty, \infty, \infty, \infty, \infty, \infty, \infty, 62, 93, 104, 170, 21, 63, 95, \infty, 23, 65, \infty, 25, \infty, \infty]$  for 6 nodes problem is obtained.

The comparison of these two scales of problems is shown in table 5 taking reasonable parameters of NP(Number of Populations), NG(Number of Generations), PC(Probability of Crossover), PM(Probability of Mutation) obtained by simulation.

Scale of the problem	NP	NG	PC	PM	Best fitness	Best rate	T(s)
$10^{18}$ (4- node)	200	150	1.0	0.3	892611.49	0.92	21
$10^{40}$ (6-ode)	200	150	1.0	0.3	890379.75	0.86	39

Table 5. Comparison of the results of different scale of problems for GA

Table 5 showed that, after the scale of the 6-node problem has expanded by  $10^{22}$  times as compared with a 4-node problem, the CPU time is increased by 18S and the best rate is decreased by 6%. Though there are expanded complexities of the problem, the algorithm still possesses high best rate. The increase in the CPU time is within an acceptable range. Moreover, time increase is mainly caused by the influences of the expansion of scale of the problem on simulation. Thus, the performance of the algorithm is not greatly affected by the scale of the problem and it is still a fairly stable algorithm.

### 3.2.2 Random-PSO algorithm

This section shows the solutions of the above model for single objective optimal control strategy problem by Random-PSO algorithm.

PSO algorithm is bring forward by Eberhart and Kennedy in 1995<sup>[17, 18]</sup>. Originally, PSO algorithm was proposed to simulate the movement of bird swarm. People observed animal society behavior, found that in a group information share was propitious to evolvement <sup>[19]</sup>, that is the basic of PSO algorithm.

PSO is based on group intelligence, its unit is the swarm, then establish simple rules for each unit, so that the whole swarm could have complex characters for solving complex optimal problems. Because of the simple concept and easily to realize, PSO develop quickly in short time, soon recognized by international evolvement calculation field, and applied in many field like electric power optimization, TSP optimization, neural networks training, digital circuit optimization, function optimization, traffic accident exploration, parameter identification.

Classical PSO optimal algorithm described as follow:

Suppose the search space is D dimension, the position of the  $i$ th particle in particle swarm is expressed as  $X_i = (X_{i1}, X_{i2}, \dots, X_{iD})$ , the speed of the  $i$ th particle is expressed as  $V_i = (V_{i1}, V_{i2}, \dots, V_{iD})$ , the best position of  $i$ th particle searched so far is denoted as  $P_i = (P_{i1}, P_{i2}, \dots, P_{iD})$ , the best position of the whole swarm have searched so far is denoted as  $P_g = (P_{g1}, P_{g2}, \dots, P_{gD})$ . For every particle, the d dimension ( $1 \leq d \leq D$ ) changes according to the equation as follow [20]:

$$V_{id} = wV_{id} + c_1r_1(P_{id} - X_{id}) + c_2r_2(P_{gd} - X_{id}) \quad (14)$$

In equation (14),  $V_{id}$  denotes the speed of the  $i$ th particle at d dimension, here: w is inertia weight, c1 and c2 are acceleration constants, r1 and r2 are random number in [0, 1] used to adjust the relative importance of  $P_{id}$  and  $P_{gd}$ , so that could obtain the next movement position of the particle:

$$X_{id} = X_{id} + V_{id} \quad (15)$$

The first part of the equation (14) is the former speed of the particle; the second part is "cognition", express the think of the particle; the third part is "social", express the information share and the cooperation between the particle [21]. "Cognition" part is explained by "law of effect" of Thorndike[22]. It is a fortified random action will possibly appear in future. The action here is "cognition", and we suppose that getting correct knowledge is enhanced, this model supposes that the particle is inspired to reduce deviation. "Social" part is explained by vicarious fortified of Bandura[23]. According to the anticipation of this theory, when the observer observe a model intensifying an action that will increase the probability of this action coming, that means the particle 's cognition will be imitated by other particle. According to equation (14) and (15) to iterate, finally obtain the optimum solution of the problem.

### 3.2.2.1 Coding

Integer coding is adopted according to the characteristic of the problem as show in section 3.2.1.1.

For describing the problem easily, we change the coding into string, the unit of solution is denoted as:  $X = (x_1, x_2, \dots, x_m)$ , here m is the length of the string,  $m = n(n+1)/2$ ; every element  $x_i$  in vector X correspond to the element of the upper triangular matrix of K,  $x_1 = K_{11}$ ,  $x_2 = K_{12}$ ,  $x_3 = K_{22}$ ,  $\dots$ ,  $x_i = K_{il}$ ,  $x_m = K_{mm}$  ( $h \leq l$ ).  $x_i \in V_i$ ,  $V_i = (x_{i1}, x_{i2}, \dots, x_{ih}) = [1, K_{ih}^u + 1]$ .  $V_i$  is the space of the  $i$ th bit of gene,  $k_i$  denote the size of this space.

### 3.2.2.2 Fitness Function

Due to the minimizing property of the objective function, fitness function is defined as the one in section 3.2.1.2.

### 3.2.1.3 Random-PSO algorithm design

Classical PSO algorithm is an effective method for searching continuous function extreme, but the research in discrete field is few. In 1997, Kennedy, Eberhart proposed "a discrete

binary version of the particle swarm algorithm", namely PSO-SV algorithm, it used to solve binary space optimal problems, that first start to utilize PSO to solve the discrete problems [21,24]. This method can only solve the binary space optimal problems, though the performance of the algorithm is excellent, its application area is restricted, for many-dimensions discrete space optimal problems, it is nail-biting. Take the problem's particularity of account, here adopt a Random-PSO algorithm[25], and use it to solve the combinatorial optimization of actual circulating cards, which is the fixing of the circulating card number in every node enterprise of supply chain inventory control strategy. The standby card number's range of every unit of the solution constitute the local search space, the global search space is consisted by  $K_{ij}^u + 1$  local search space ( $K_{ij}^u$  is the digit capacity of the solution), that is all the standby card number's range constitute global search space. The structure of solution of the problem is shown in figure 4[25].

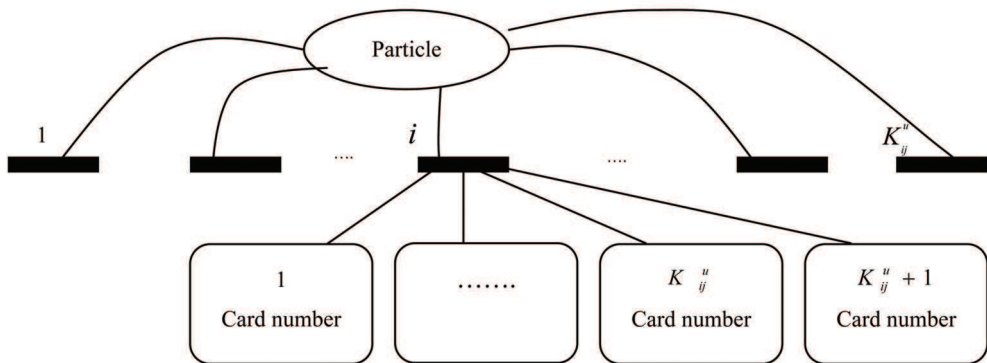


Fig. 4. Structural diagram of solution space

Every particle denotes the whole solution of the problem. The solution is consisted of three levels, first level is particle level, second level is every unit of solution, third level is card number; the card number of every unit constitute a local search space, the particle firstly search in the local space, choose a card number for every unit, then the card number at all unit constitute a solution. It is easy to see that the card number at all units constitute the global search space.

The speed and position of the particle update as follow[25]:

$$V_{id_j} = r \cdot \dot{V}_{id_j} + r_1 \cdot \dot{(P_{id} - X_{id_j})} + r_2 \cdot \dot{(P_{gd} - X_{id_j})}$$

$$= \begin{cases} r \cdot \dot{V}_{id_j} + (p_{id} - X_{id_j}) + (p_{gd} - X_{id_j}) & r_1 > 0.5, r_2 > 0.5 \\ r \cdot \dot{V}_{id_j} + (p_{id} - X_{id_j}) & r_1 > 0.5, r_2 \leq 0.5 \\ r \cdot \dot{V}_{id_j} + (p_{gd} - X_{id_j}) & r_1 \leq 0.5, r_2 > 0.5 \\ r \cdot \dot{V}_{id_j} & r_1 \leq 0.5, r_2 \leq 0.5 \end{cases} \quad (16)$$

$$X_{id_j} = \begin{cases} 0 & X_{id_j} < 0 \\ J_d & X_{id_j} > J_d \\ X_{id_j} + V_{id_j} & 0 \leq X_{id_j} \leq J_d \end{cases} \quad (17)$$

$$num_{id_j} = \begin{cases} num_{id_j} + 1 & X_{id_j} = j \\ num_{id_j} & X_{id_j} \neq j \end{cases} \quad (18)$$

$$F_{id_j} = \frac{num_{id_j}}{J_d + 1} \quad (19)$$

The normalization of  $F_{id_j}$  :

$$sum_{id} = \sum_{j=0}^{J_d} F_{id_j} \quad (20)$$

$$P_{id_j} = \frac{F_{id_j}}{sum_{id}} \quad (21)$$

To generate a random number in 0-1 for every unit of every particle, denote as rand,

$$X_{id} = \arg_j (P_{id_j} < rand \leq P_{id_{j+1}}) \quad (22)$$

$X_{id}$  is the code of the circulating card chosen by the  $d$ th unit of the  $i$ th particle. Here “ $\bullet$ ” is different from the normal product, it is a binocular operator, two parts of their operands can not reverse, the former part is a random number control the effect of the other one which is a integer:  $V_{id} \in [-2J_d, 2J_d]$ ,  $P_{id} P_{gd} X_{id_j} \in \{0, \dots, J_d\}$ ;  $r \in (0, 1)$  is inertia factor, used to adjust the speed,  $r_1, r_2 \in (0, 1)$  are random numbers, used to adjust the extreme of particle and the global extreme.;  $num_{id_j}$  note the times of card number which is  $j$  at the  $d$ th unit of the  $i$ th particle,  $P_{id_j}$  is the frequency of card number which is  $j$  at the  $d$ th unit of the  $i$ th particle, the probability is bigger as this value for the card number being  $j$ .

#### 3.2.1.4 The Procedure for Random-PSO

The main procedure for Random-PSO is as follows<sup>[25]</sup>:

Step1: NC  $\leftarrow$  0 (NC is iterative number)

To produce a random number  $j$  from every unit of particle  $i$ ,  $j \in \{0, \dots, J_i\}$ ,  $X_{id_j} = j$  constitute the initial position of the particle, equation (19)-(20) produce the initial solution, assign this value to particle extreme, take the better one as the global extreme; Let the initial speed  $v=0$ ;

Step 2: if get the maximal iterative number, go to step 7, else go to step 3;

Step 3: use the control matrix  $K$  as the parameter to call the simulation, obtain three economic indexes, and calculate the fitness value.

Step 4: compare the currently particle fitness value and the particle extreme for every particle  $i$ , if the currently particle fitness value is better, then update  $P_{id}$ ;

Step 5: compare the currently particle fitness value and the global extreme, if the currently particle fitness value is better, then update  $P_{gd}$ ;

Step 6: update the  $V_{id_j}$  and  $X_{id_j}$  follow equation (16) and (17), and produce a new solution from equation (18)-(22), go to step 2;

Step 7: output the optimal objective function value and the card combination.

### 3.2.1.5 Numerical Analysis

In order to test the efficiency of the random-PSO algorithm, two problems in section 3.2.1.5 is used here. The comparison of the two problems is shown in table 6 taking reasonable parameters of NP and NG obtained by simulation.

Problem scale	NP	NC	Best fitness	Best rate	T(s)
$10^{18}$ (4 nodes)	150	100	892611.49	0.94	13
$10^{40}$ (6 nodes)	150	150	890379.75	0.90	54

Table 6. Comparison of the results of different scale of problems for random\_PSO

Table 6 showed that, after the scale of the 6-node problem has expanded by  $10^{22}$  times as compared with a 4-node problem, the CPU time is increased by 41s and the optimization percentage is decreased by 4%. Though there is the expanded complexity of the problem, the algorithm still possesses a better optimization percentage. The increase in the CPU time is within an acceptable range. Moreover, time increase is mainly caused by the influences of the expansion of scale of the problem on simulation. Thus, conclusively speaking, the optimal performance of the algorithm is not greatly affected by the scale increase of the problem.

### 3.3.3 Pheromone evolutionary algorithm

This section shows the solutions of the above model for single objective optimal control strategy problem by PEA.

Huang et al.<sup>[26]</sup> propose the evolutionary algorithm based on pheromone. It is a process of probability choices by making use of pheromone, which is one of the important concepts in the algorithm of ant system<sup>[27]</sup>. In the ant system, pheromone is evenly distributed in the solution space, and through the positive feedback of pheromone, the algorithm is converged to the optimal point of the whole set. In the discrete problem of evolutionary algorithm, the mutation operators every time undergo only one or several emergence in gene positions.

These mutation operators are lack of direction and the mutation is not even. Therefore the ability for searching optimization is weak. If pheromone is induced into the mutation operators in every mutation, the mutation operators will be more directed by making use of the changes in pheromone, which is strongly related to the fitness value. These changing pheromone guides the mutation of each gene position. Thus the mutation operators raise greatly their abilities for searching optimization. With the help of the mutation operators, the algorithm can steadily converged to the optimal of the whole search space.

The evolutionary algorithm combines the concept of pheromone and the mutation operators of genetic algorithm. The probability field represents pheromone. A series of code that represents the solution of the problem is used to represent the gene series. Same probability is given to each bit of the gene in the series for initiation. Then another series is created randomly from these probabilities and their fitness values are calculated. These fitness values help to adjust the distribution of chosen probabilities. The above process is repeated until the probability distribution is steady. The series that combines the possible values of the greatest chosen probabilities is the final solution. Here, through adjustment of the probability field, each gene position of the gene series undergoes mutation and each gene can inherit their fathers' characteristics. Under the operation of directional mutations, the algorithm is converged finally.

### 3.3.3.1 Coding

Integer coding is adopted according to the characteristic of the problem as show in section 3.2.1.1.

Following this coding rule, each solution unit has different upper bound of card number according to the numbers of nodes it has. For the sake of convenience, the above coding is changed into serial coding. Each solution unit can be described as:  $X = (x_1, x_2, \dots, x_m)$  where  $m$  is the length of the series and  $m = n(n+1)/2$ . Each element  $x_i$  of vector  $X$  corresponds to the element of the upper triangular part of matrix  $K$ , that is,

$$x_1 = K_{mm}, x_2 = K_{12}, x_3 = K_{22}, \dots, x_i = K_{hl}, x_m = K_{mm} (h \leq l)$$

$x_i \in V_i, V_i = (x_{i1}, x_{i2}, \dots, x_{ik_i}) = [1, K_{hl}^u + 1]$ .  $V_i$  is the space of the  $i$ th bit of gene,  $k_i$  denote the size of this space.

$V_i$  corresponds to a probability distribution.  $P_i = (p_{i1}, p_{i2}, \dots, p_{ik_i})$  where  $p_{ij}$  corresponds to  $x_{ij}$  and  $\sum_{j=1}^{k_i} p_{ij} = 1$ .  $P_i$  is the probability field of  $V_i$  and  $P = \{P_i\}$ .

### 3.3.3.2 Fitness function

Due to the minimizing property of the objective function, fitness function is defined as the one in section 3.2.1.2.

### 3.3.3.3 Generating new generation

The value of  $X$  is formed according to the probability field and  $x_i$  is chosen by roulette method[26].

$$\text{If } \sum_{j=1}^{i-1} P_j < \text{rand}[0,1] \leq \sum_{j=1}^i P_j, \text{ then } x_i = x_{ij} \quad (23)$$



Where  $P_{ij} = \sum_{h=1}^j p_{ih}$ ,  $P_{i0} = 0$ . The values of  $x_i (i = 1, 2, 3, \dots, m)$  are obtained by the above method. Thus  $X$  is formed.

#### 3.3.3.4 Adjustment of the probability field

In order to let the probability field converge, continuous adjustment is needed. Assume that the fitness value of a certain solution unit  $X_{old}$  of the last generation is  $fitness_{old} = F(X_{old})$  where  $F(X)$  is a function of fitness value that is determined by the practical problem.

When the fitness value of the unit  $X_{new}$  of the preceding generation  $fitness_{new} = F(X_{new})$ , the probability field  $P$  of  $x_{ij}$  is adjusted as below:

The probabilities  $p_{ij}$  that correspond to  $x_{ij}$  are changed into  $p_{ij} \cdot (1 + \Delta p_{ij})$ . Then  $p_{i1}, p_{i2} \dots p_{ik_i}$  undergo the unified process and  $i = 1, 2, 3, \dots, m$ . Here,

$$\Delta p_{ij} = \alpha \cdot \arctan \left( \frac{fitness_{new} - fitness_{old}}{fitness_{old}} \right) \quad (24)$$

$\Delta p_{ij}$  is the amount for probability adjustment.  $\alpha$  is the adjustment coefficient. The choices of adjustment coefficient of the probability field and the stoppage parameters are important and delicate. If the adjustment coefficient of the field is too big, then the speed of convergence is too fast, it will easily give the local optimization. If the coefficient is too small, then the speed is too slow, and the efficiency of the algorithm becomes low. That is why the substantial adjustment has to be done according to the practical problem of the instance simulation. The continuous adjustment of  $P$  increases the probability of getting solution units with high fitness values. At the same time, through unified process, the probability of having solution units with low fitness values decreases. This situation gives a very high probability of getting the optimal solution. The function in probability adjustment is a tangent function,  $\arctan(x)$ . It prevents the value of  $\Delta p_{ij}$  being too large to find the global optimal solution.

Actually, in the adjustment process, the fitness values are the directed values that operate mutation in the probability field. Mutations happen at each gene position of every generation simultaneously. Therefore, the genes have learned their fathers' characteristics thoroughly and this hastens the algorithm's speed of convergence.

#### 3.3.3.5 Initialization

In order to guarantee the algorithm searches within global space, at the initiation stage, each solution should have the same probability of being chosen. That is  $p_{ij}$  which should have even distribution.

$$p_{ij} = \frac{1}{k_i}, I = 1, 2, 3, \dots, m, j = 1, 2, 3, \dots, k_i \quad (25)$$

#### 3.3.3.6 The termination criterion

The largest generation is chosen as the termination criterion.

#### 3.3.3.7 The Procedure for PEA

The main steps in the flow of the solution are:

Step 1:  $NC \leftarrow 0$  (NC are the iterative steps)

The algorithm based on pheromone evolution produces the population  $X$  of the initial solutions and their corresponding initial probability field  $P$ . The actual procedure is to create a random number  $q$  for each  $x_{ij}$  of every solution unit  $x_i$ .  $q \in V_i$  and  $x_{ij} = q$  forms the initial solution  $X$  which becomes the input for the simulation. The probability field is initialized according to formula (25);

Step 2: The fitness values  $fitness_{old} = F(X_{old})$  of each solution of the last generation  $X_{old}$  are calculated; then turn to step 3;

Step 3: If the termination criterion is satisfied, turn to step 6; or else, create a solution of the new generation  $X_{new}$  and process it in the simulation. Then the fitness values  $fitness_{new} = F(X_{new})$  of each solution of the present generation are calculated;

Step 4: The new probability field that corresponds to  $X_{new}$  is obtained by adjusting the probability field according to formula (24);

Step 5: Let  $X_{old} \leftarrow X_{new}$ , then turn to step 2;

Step 6: The optimal value of the objective function and the corresponding circulating card number are output.

### 3.3.3.8 Numerical Analysis

In order to test the efficiency of the PEA algorithm, two problems in section 3.2.1.5 is used here. The comparison of the two problems is shown in table 7 taking reasonable parameters of NP, NG, PC, PM obtained by simulation.

Problem scale	NP	NG	Best fitness	Best rate	T(s)
$10^{18}$ (4 nodes)	200	150	892611.49	0.92	19
$10^{40}$ (6 nodes)	200	150	890379.75	0.88	41

Table 7. Comparison of the results of different scale of problems for PEA

Table 7 has shown that, after the scale of the 6-node problem has been expanded by  $10^{22}$  times as compared with a 4-node problem, the CPU time is increased by 22s and the optimization rate is decreased by 4%. Though there is the expanded complexity of the problem, the algorithm still possesses a better optimization rate and the increase in the CPU time is within an acceptable range. Moreover, time increase is mainly caused by the influences of the expansion of scale of the problem on simulation. Thus, conclusively speaking, the optimal performance of the algorithm is not greatly affected and it is still a fairly stable algorithm.

### 3.3.4 The comparison among the different algorithms

Finally the GA, Random-PSO and PEA is compared. The results are shown in Table 8.

Algorithm (problem scale)	NP	NG	Best rate	T (s)
GA ( $10^{18}$ )	200	150	0.92	21
Random-PSO ( $10^{18}$ )	150	100	0.94	13
PEA ( $10^{18}$ )	200	150	0.92	19
GA ( $10^{40}$ )	200	150	0.86	39
Random-PSO ( $10^{40}$ )	150	150	0.90	54
PEA ( $10^{40}$ )	200	150	0.88	41

Table 8. Comparison among GA, Random-PSO and PEA for different scale of problems

Table 8 has shown that, the Random\_PSO has highest optimization rate when the scale of problem increase, and the CPU time of the three algorithms is similar. Therefore, the Random-PSO is more effective than GA and PEA for this kind of problem.

### 3.4 Conclusions

Determination of the optimal control strategy is a key factor for a successful supply chain. This chapter has made researches on the optimal inventory control strategy of a serial supply chain and presents the description of a two-level model. The first-level model is a nonlinear integer-programming model. Its main purpose is to determine the optimal control strategy which gives the minimal overall cost of a supply chain under some constraints. These constraints include the customer service no less than the given value and the standard deviation of input less than a given value. When the inventory control strategy is given, the second-level model is used to obtain the performance measurements of the supply chain. The first-level model reaches optimization through the algorithm based on intelligent algorithm. The intelligent algorithms of GA, random-PSO and PEA is considered in this study. The second-level model implements the general push/pull model of inventory of a serial supply chain by simulation. The main characteristic of the second-level model is that, the choice of control, push or pull, of a node is determined by whether it is under the feedback control of its downstream nodes. This has the potential to be an efficient quantitative tool for more complex SC analysis in the global business environment.

Instances of different scales of problems are analyzed. The results shown the effectiveness and the efficiency of the method. The cost of the whole supply chain is minimized while satisfying the customer demands and limiting the "bullwhip effect". It balances production rhythm and shared benefits of each node enterprise of a supply chain, which gives a quantitative support of rational organization of purchase, production, transportation, and sales.

Finally, comparison study of different scales of problem for three intelligent algorithms is given. Results shows that the random\_PSO has highest optimization rate when the scale of problem increase, and the CPU time of the three algorithms is similar. It suggested that the random-PSO is more effective than the other two algorithm for this kind of problem.

This work was supported by the National Natural Science Foundation of China (Grant no. 70671020, 70721001, 70431003, 60673159), the Program for New Century Excellent Talents in University (Grant no. NCET-05-0295, NCET-05-0289), Specialized Research Fund for the Doctoral Program of Higher Education (Grant no. 20070145017, 20060145012), the National High-Tech Research and Development Plan of China (Grant No. 2006AA01Z214), the Key Project of Chinese Ministry of Education (Grant No. 108040), the Natural Science Foundation of Liaoning Province (Grant No. 20062022).

### 4. Reference

- Chopra S, Meindl P, Supply Chain Management (3rd Edition) , Prentice Hall, 2006
- Hugos M, Essentials of Supply Chain Management (2nd Edition), Wiley , 2006.
- Dolgui A, Soldek J, Zaikin O, Supply chain optimization: product/process design, facility location and flow control, New York: Springer, 2005.
- Gstettner S, Kuhn H, Analysis of Production Control Systems Kanban and CONWIP. International Journal of Production Research, 1996, 34(11): 3253-3273.
- Deleersnyder J. L., Hodgson T. J., Mueller H, et al. Kanban Control Led Pull System: Analysis Approach. Management Science, 1989, 35(9): 1079-1091.

- Sarker B. R., Fitzsimmons J. A., The Performance of Push and Pull Systems: A Simulation and Comparative Study. *International Journal of Production Research*, 1989, 27(11): 1715-1732.
- Flapper S. D. P., Miltenburg G. J., Wijugard J., Embedding JIT into MRP. *International Journal of Production Research*, 1991, 29(2): 329-341
- Larsen N. E. Altung L. Criteria for selecting production control philosophy. *Production Planning & Control*, 1993, 4(1): 54-68
- Villa A., Watanabe T. Production management: Beyond the Dichotomy Between "Push" and "Pull". *Computer Integrated Manufacturing Systems*, 1993, 6(1): 53-63
- Spearman, M. L., Woodruff, D. L., and Hopp, W. J. CONWIP: a Pull Alternative to Kanban. *International Journal of Production Research*, 1990, 28(5): 879-894.
- Gaury E, Kleijnen JPC and Pierreval H, A methodology to customize pull control systems, *Journal of the Operational Research Society*, 2001, 52, 789-799.
- Ovalle O R., Marquez A C., Exploring the Utilization of a CONWIP System for Supply Chain Management. *International Journal of Production Economics*, 2003, 83:195-215
- Herer Y. T., Masin M. Mathematical Programming Formulation of CONWIP Based Production Lines and Relationships to MRP. *International Journal of Production Research*, 1997, 35 (4): 1067-1076.
- Bose, G.J., Rao, A. Implementing JIT with MRP Creates Hybrid Manufacturing Environment. *Industrial Engineering*. 1988, 20(9): 49-53.
- Michalewicz, M, Genetic algorithms + data structure = evolution programs. (3rd rev. and extended ed). Berlin; Hong Kong: Springer, 1996
- Mitchell, M. An introduction to genetic algorithms. The MIT Press, Cambridge, Massachusetts, London, England, 1996.
- Kennedy J , Eberhart R. Particle swarm optimization. *Proc IEEE International Conference on Neural Networks*, Perth, Australia 1995, 1942-1948.
- Eberhart R C, Kennedy J A. A new optimizer using particle swarm theory. *Proc 6th Int. Symposium on Micro Machine and Human Science*. Nagoya, 1995. 39-43.
- Wilson E O. *Sociobiology : The New Synthesis*. .MA: Belknap Press, 1975.
- Shi Y., Eberhart R. A modified particle swarm optimizer. *Proc IEEE Int. Conf Evolutionary Computation*. Anchorage, 1998. 69-73.
- Kennedy J. The particle swarm: Social adaptation of knowledge. *Proc IEEE Int. Conf on Evolutionary Computation*. Indian apolis, 1997. 303-308.
- Thorndike E L. *Animal Intelligence: Empirical Studies*. New York: MacMillan, 1911.
- Bandura A .*Social Foundations of Thought and Action :A Social Cognitive Theory*. New Jersey: Prentice 2 Hall, 1986.
- Kennedy J. Eberhart R.A Discrete binary version of the particle swarm algorithm. In: *Proceedings of the IEEE International Conference on Systems. Man and Cybernetics*, 1997, 5:4104-4108.
- Huang M, Wu X J, Wang X W, Ip W X, Yung K L, The risk programming of virtual enterprises based on algorithms of Min / Max / Mean / Random-PSO, 1st IEEE Conference on Industrial Electronics and Applications (ICIEA 2006), 2006.05, Singapore, 997-100
- Huang Z. C., Wu F. C., Hu X. L., An evolutionary algorithm to integer programming problem based on pheromone, *Research on Computer Applying*, 2001, 7(2), 27-29.
- Dorigo M, Maniezzo V. A colony ant system: optimization by a colony of cooperating agent, *IEEE Tran. On System, Man and Cybernetics Part B*, 1996, 26(1), 34-37.



## **Supply Chain the Way to Flat Organisation**

Edited by Julio Ponce and Adem Karahoca

ISBN 978-953-7619-35-0

Hard cover, 436 pages

**Publisher** InTech

**Published online** 01, January, 2009

**Published in print edition** January, 2009

With the ever-increasing levels of volatility in demand and more and more turbulent market conditions, there is a growing acceptance that individual businesses can no longer compete as stand-alone entities but rather as supply chains. Supply chain management (SCM) has been both an emergent field of practice and an academic domain to help firms satisfy customer needs more responsively with improved quality, reduction cost and higher flexibility. This book discusses some of the latest development and findings addressing a number of key areas of aspect of supply chain management, including the application and development ICT and the RFID technique in SCM, SCM modeling and control, and number of emerging trends and issues.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Min Huang, W.H.Ip, Xingwei Wang and Jianqing Ding (2009). Optimal Control Strategy for Serial Supply Chain, Supply Chain the Way to Flat Organisation, Julio Ponce and Adem Karahoca (Ed.), ISBN: 978-953-7619-35-0, InTech, Available from:

[http://www.intechopen.com/books/supply\\_chain\\_the\\_way\\_to\\_flat\\_organisation/optimal\\_control\\_strategy\\_for\\_serial\\_supply\\_chain](http://www.intechopen.com/books/supply_chain_the_way_to_flat_organisation/optimal_control_strategy_for_serial_supply_chain)

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.