

## Hybrid Particle Swarm Optimization for Two-stage Cross Docking Scheduling

Hairu Zhao and Ling Chen

College of Automation, Chongqing University, Chongqing, 400030, China  
[cchenlingg@163.com](mailto:cchenlingg@163.com)

### Abstract

*In order to improve supply chain's operation efficiency, shorten delivery time and decrease distribution costs, two-stage cross docking scheduling problem under direct shipment mode was studied in this paper. Taking into consideration the influence of numbers of vehicles in distribution center on cross docking problem, three models were established based on different assumptions including only one vehicle in distribution center, many vehicles in distribution center and the location of distribution center to be determined, and the objective was to minimize transportation time. Hybrid particle swarm optimization was proposed to solve the model on the basis of PSO and GA. The algorithm introduced clone selection operator to make particles multiply and mutate by calculating the affinity between individuals so that the best individual can be reserved and the poor can be improved. Clone operator, crossover operator, antibody reorganization operator and mutation operator were designed to improve the performance of the algorithm. Computational experiments showed that the hybrid particle swarm optimization algorithm has faster convergence speed and better solution precision compared with other algorithms. The result of the present work implied that the model in this paper was accord with the reality, and it was effective and feasible.*

**Keywords:** cross docking; vehicle scheduling; hybrid particle swarm optimization

### 1. Introduction

Cross docking operation is a process that the products are directly sorted, integrated then sent to the outbound door according to customer's demand only when delivered from supplier to the distribution center. That is to say, cross docking is an operation that products stay little time in warehouse. Cross docking brings significant benefits including little or no inventory, low handling costs, low space requirement, centralized processing and low transportation costs. It is one of the major contributions to improve supply chain performance, and plays an important role in affecting supply chain's operation efficiency. So it's necessary to study cross docking scheduling problem.

Due to the success and the development of cross docking during the past years, several papers can be found in the literature about problems related to cross-docking management.

The cross docking scheduling model has been first proposed by Chen and Lee [1] from the scheduling points of view. They study the problem in which a job at the second machine can be processed only after the processing of some jobs at the first machine has been completed. The problem is denoted as F2|CD|Cmax and the objective is to minimize the makespan. They show the problem is NP-hard in strong sense, and develop approximation algorithm and branch and bound algorithm based on the several characteristics of its feasible solution. This model has been extended to multi-docks by Chen and Song [2]. This extension is denoted as HF2 (m1, m2)|CD|Cmax where HF means hybrid flow shop, and m1 and m2 are the number of machines in stages 1 and 2, respectively. Ma and Chen [3] also study the cross docking scheduling problem with total

completion time, a dynamic programming is designed with computational complexity of  $O(nm^2m)$ .

Most papers study cross docking scheduling based on different assumptions. Yu and Egbelu [4] as well as Boysen, *et al.*, [5] deal with the truck scheduling assuming that there is merely a single inbound and a single outbound door in cross docking center. While Jan Van Belle, *et al.*, [6] present a truck scheduling problem that is concerned with both inbound and outbound trucks at multiple dock doors. Shakeri, *et al.*, [7] pay attention to truck scheduling in a resource-constrained cross docking. Boysen [8] treats a special truck scheduling problem arising in the (zero-inventory) cross docks of the food industry, where strict cooling requirements forbid an intermediate storage inside the terminal, so that all products are to be instantaneously loaded onto refrigerated outbound trucks. Rim Larbi, *et al.*, [9] give attention to the transshipment scheduling problem in a single receiving and a single shipping door cross dock under three scheduling policies. They assume to have complete information on the order of arrivals and the contents of all inbound trucks, in which, an optimal graph based model is proposed and a polynomial time algorithm is given. They also assume the availability of partial and no information on the sequence of upcoming trucks, in which, heuristics are developed for the two cases.

According to various considerations, researchers build different models. Yu and Egbelu [4] build an integer programming (IP) model with the objective to find the best truck docking or scheduling sequence for both inbound and outbound trucks to minimize total operation time when a temporary storage buffer to hold items temporarily is located at the shipping dock. While Musa and Arnaout, *et al.*, [10] use an integer programming model with the objective to minimize the transportation cost in a cross docking network. Bellanger, *et al.*, [11] present a three-stage hybrid flow shop model, in which shipments and orders are represented as batches. The first stage corresponds to the receiving docks, the second stage corresponds to the sorting stations, and the third stage corresponds to the shipping docks. Mousavi, *et al.*, [12] present a two-stage mixed-integer programming (MIP) model for the location of cross docking centers and vehicle routing scheduling problems with cross docking due to potential applications in the distribution networks. Arabani, *et al.*, [13] establish a multi-objective model for cross docking scheduling problem. Objective functions are considered as the total operational time (makespan) and the total lateness of all outbound trailers. Dong and Tang, *et al.*, [14] address a vehicle routing and scheduling problem arising in Flight Ticket Sales Companies for the service of free pickup and delivery of airline passengers to the airport. A 0–1 mixed integer programming model is presented, in which service quality is factored in constraints by introducing passenger satisfaction degree functions that limit time deviations between actual and desired delivery times.

In order to solve cross docking scheduling problem effectively, all kinds of methods are proposed and adopted. Jan Van Belle, *et al.*, [6] present a tabu search (TS) approach to solve the problem, and the results indicate that the proposed tabu search is able to find good quality results in a short time period. Santos, *et al.*, [15] discuss a Branch-and-price algorithm for the Pickup and Delivery Problem with Cross-Docking (PDPCD). The computational results indicate that optimal or near optimal solutions for PDPCD indeed allow total costs to be significantly reduced and the Branch-and-price algorithm for PDPCD works better than similar algorithms for other models in the literature. Musa and Arnaout, *et al.*, [10] use ant colony optimization (ACO) algorithm to solve the transportation problem of cross docking network. The results show that their proposed approach found solutions that significantly reduce the shipping cost in the network of cross-docks and considerably outperform Branch-and-Bound algorithm especially for large problems. Arabani, *et al.*, [13] address three famous multi-objective algorithms including non-dominated sorting genetic algorithm-II (NSGA-II), strength Pareto evolutionary algorithm-II (SPEA-II), and sub-population genetic algorithm-II (SPGA-II) to solve the cross docking scheduling problem. Vahdani and Zandieh [16] apply five

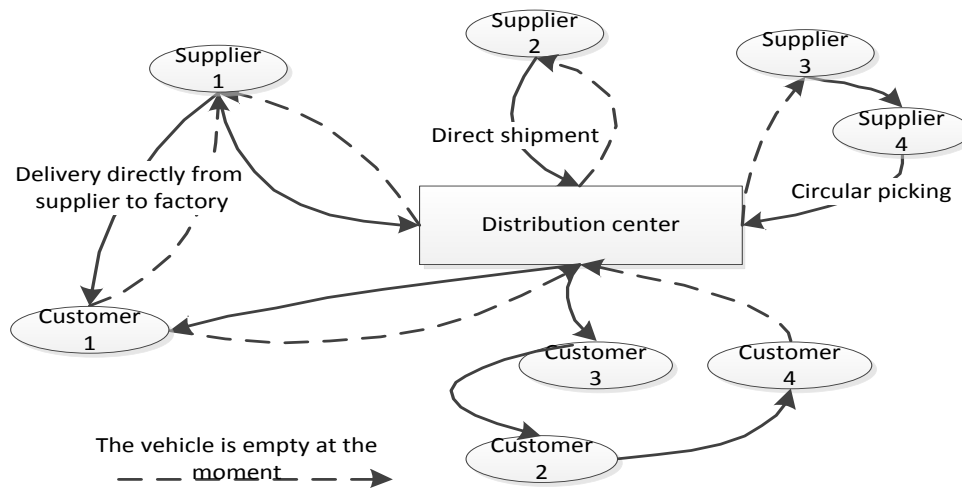
meta-heuristic algorithms: genetic algorithm (GA), tabu search (TS), simulated annealing (SA), electromagnetism-like algorithm (EMA) and variable neighborhood search (VNS) to schedule the trucks in cross-dock systems. Arabani and Ghomi, *et al.*, [17] analyze the characteristics of five heuristic algorithms for cross docking problem including genetic algorithm (GA), tabu search (TS), particle swarm optimization (PSO), ant colony optimization (ACO) and differential evolution (DE). Liao, *et al.*, [18] solve the scheduling problem by three meta-heuristic algorithms, which include differential evolution, DE-AGZ algorithm and HDE. He extends the meta-heuristic algorithm to six [19], which include simulated annealing, tabu search, ant colony optimization, differential evolution, and two hybrid differential-evolution algorithms. Vahdani, *et al.*, [20] propose a new hybrid meta-heuristic for vehicle routing scheduling in cross-docking systems. This new hybrid algorithm incorporates the elements from Particle Swam Optimization, Simulated Annealing and Variable Neighborhood Search to enhance its search capabilities. Wang and Lu [21] propose a novel hybrid genetic algorithm (HGA) for solving a capacitated vehicle routing problem (CVRP).

Up to now, no one consider how the numbers of vehicles in distribution center influence cross docking scheduling problem. So this paper takes account of the amount of vehicles in distribution center to describe the problem and arrange the sequence of the inbound vehicle. According to the assumption, three mathematical models are built under the direct shipment mode called a single vehicle scheduling model, vehicles scheduling model and vehicles scheduling model with the distribution center to be determined, which correspond to only one vehicle in distribution center, many vehicles in distribution center and the location of distribution center to be determined. The model's objective is to minimize transportation time. Because every algorithm isn't perfect enough. To full use of its advantage and avoid its disadvantage, particle swarm optimization and genetic algorithm are combined to propose hybrid particle swarm optimization to solve the model. Computational experiments show that the algorithm can solve the problem optimally.

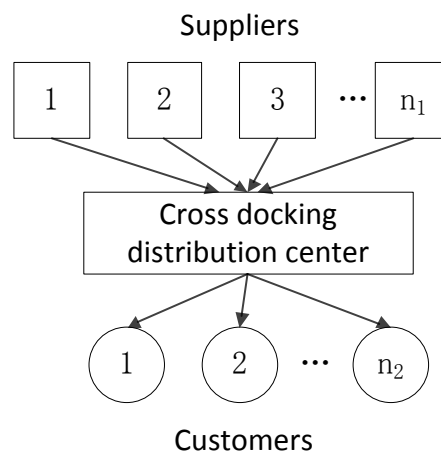
The remainder of the paper is organized as follows. In Section 2, we describe our problem more exactly and define the equivalent corresponding concepts between cross docking vehicle scheduling problem and two-stage flow shop scheduling problem. In Section 3, we establish three mathematical models called a single vehicle scheduling model, vehicles scheduling model and vehicles scheduling model with the distribution center to be determined. Hybrid particle swarm optimization is proposed to solve the model in Section 4. We present our computational experiments and results in Section 5 and the paper is closed in Section 6, where we offer some conclusions.

## 2. Problem Description

Cross docking in distribution center includes inbound logistics and outbound logistics according to the area partition. Inbound logistics (as shown in figure 1) generally adopt two kinds of transportation modes: direct shipment and the transportation with circular pick-up. Direct shipment means that a truck transports the products of one supplier or customer at a time. On the contrary, the transportation with circular picking means that a truck carries multiple suppliers' or customers' products when some suppliers' or customers' products are not enough to fill up a truck. Circular picking is suitable for multi-period frequency and small batch transportation, while direct shipment is suitable for direct transportation with loading on a large scale. This paper does not consider the transportation with circular picking, is only for direct shipment which can be found everywhere in practice. This mode greatly reduces the cycle time from an order to delivery and the damage rate of the product, and saves the costs of circulation. It is important for the products which have high requirement to the freshness and have a time limit to adopt this transportation mode.



**Figure 1. Schematic Plot of Inbound Logistics**



**Figure 2. Example of Supply Chain with Cross Docking Distribution Center**

Figure 2 shows the supply chain with cross docking distribution center. A set  $S$  of suppliers, supplies all kinds of products in the first stage, and all products are transported to the set  $V$  of distribution center by vehicles. There is a set  $D$  of customers in the second stage, and each customer has a demand subset corresponding to the set  $S$  of suppliers. The inbound sequence of products should be considered in direct shipment. An order may demand for many products from different suppliers, and one supplier's products may be needed by different orders. Certain products' arrival delay would lead other products to wait, thus it would increase transportation inventory. If an order can't be satisfied, other orders will be delayed at the same time. In a word, arranging the sequence of inbound vehicles properly is an important step in cross docking so that the operating efficiency can be improved on the premise that customers' demands are met.

This paper transforms the cross docking vehicle scheduling problem into two-stage flow shop scheduling problem. Delivery vehicles can be thought of machines, pickup and delivery time can be thought of processing time in two machines. Table 1 shows the corresponding concepts between cross docking vehicle scheduling problem and two-stage flow shop production scheduling problem (aiming at direct shipment).

**Table 1. Equivalent Corresponding Concepts**

Name in cross docking vehicle scheduling problem	Name in two-stage flow shop scheduling problem
inbound vehicle $k$	machine $k$ in the first group machines
outbound vehicle $l$	machine $l$ in the second group machines
supplier $i$	task $J_{1i}$ which needs to be processed in the first group machines
order $j$	task $J_{2j}$ which needs to be processed in the second group machines
The products of supplier $i$ are picked up by inbound vehicle $k$ .	Process task $J_{1i}$ , $J_{1i} \in J_1$
The products of order $j$ are transported by outbound vehicle $l$ .	Process task $J_{2j}$ , $J_{2j} \in J_2$
time $t_{i1}$ of picking up products from supplier $i$	processing time $t_{i1}$ of task $J_{1i}$ in the first group machines
time $t_{j2}$ of delivering the order $j$	processing time $t_{j2}$ of task $J_{2j}$ in the second group machines

Because it takes such short time to sort and integrate products in distribution center that the time is almost negligible relative to the products' transportation time. This paper only considers the transportation time  $t_{i1}$  of products from the supplier to the distribution center and the time  $t_{j2}$  of products from the distribution center to the customer when studying cross docking vehicle scheduling problem, and establishes mathematical model with the objective to minimize transportation time.

### 3. Models

#### 3.1. A Single Vehicle Scheduling Model

##### 3.1.1. Basic Assumptions:

- (1) This paper adopts the mode of direct shipment, that is to say, each supplier's products only need one vehicle to transport to the distribution center and the products from one customer's order can be delivered at one time from distribution center.
- (2) When a supplier's products are delivered to the distribution center, and then sent to the customer immediately. The integration time is not taken into account.
- (3) The number of vehicle in two stages of cross docking scheduling is 1.
- (4) There is no interruption after each item starts being shipped.

##### 3.1.2. Parameters:

- $n$  : Number of supplier in first stage
- $m$  : Number of customer in second stage
- $P_{i1}$  : Transportation time of the products of supplier  $J_{i1}$
- $P_{j2}$  : Transportation time of the products of order  $J_{j2}$
- $M$  : A number large enough

### 3.1.3. Decision Variables:

$T$  : Transportation time

$C_i$  : Starting shipping time of the products of supplier  $J_{i1}$

$D_j$  : Starting shipping time of the products of order  $J_{j2}$

$$g_{ii'} = \begin{cases} 1, & \text{if task } J_{i1} \text{ is handled before task } J_{i'1} \\ 0, & \text{otherwise} \end{cases}$$

$$h_{jj'} = \begin{cases} 1, & \text{if task } J_{j2} \text{ is handled before task } J_{j'2} \\ 0, & \text{otherwise} \end{cases}$$

### 3.1.4. Model I:

$$\min T \tag{1}$$

s.t.

$$T \geq D_j + P_{j2}, j = 1, \dots, m \tag{2}$$

$$C_i \geq C_i + P_{i1} - M g_{ii'}, i = 1, \dots, n, i' = 1, \dots, n \tag{3}$$

$$C_i \geq C_i + P_{i1} - M (1 - g_{ii'}), i = 1, \dots, n, i' = 1, \dots, n \tag{4}$$

$$D_j \geq D_j + P_{j2} - M h_{jj'}, j = 1, \dots, m, j' = 1, \dots, m \tag{5}$$

$$D_j \geq D_j + P_{j2} - M (1 - h_{jj'}), j = 1, \dots, m, j' = 1, \dots, m \tag{6}$$

$$D_j \geq C_i + P_{i1}, i = 1, \dots, n, j = 1, \dots, m \tag{7}$$

$$C_i \geq 0, D_j \geq 0, i = 1, \dots, n, j = 1, \dots, m \tag{8}$$

Objective function (1) minimizes the transportation time. Constraint (2) ensures maximum transportation time is larger than or equal to the time of products delivered from distribution center to customer in the second stage. Constraints (3) and (4) ensure only one supplier's products are transported in the first stage at the same time. Constraints (5) and (6) ensure only one customer's products are transported in the second stage at the same time. Constraint (7) ensures only when the products of a certain order are all delivered in the phase 1, the corresponding order's products will be transported in phase 2. Constraint (8) enforces the starting time nonnegative.  $T$  is the optimal solution under a certain situation by solving the above model, and the corresponding values of  $g_{ii'}$  and  $h_{jj'}$  are the optimal ordering strategy.

### 3.2. Vehicles Scheduling Model

In reality, there is more than one truck in distribution center, thus vehicles scheduling model is put forward in which there are many trucks in every phase and these trucks are same. The model can be described as follows: suppliers' products are delivered to the distribution center by multiple vehicles in stage 1. After simply split, combined and sorted, these products are transported to customers by multiple vehicles in stage 2. It's worth to note that customers' demands may come from different suppliers and the products from multiple suppliers may be transported by different trucks. So the products can't be integrated and sent to customers until all products of an order arrive in distribution center. That is to say, if a certain order can't be split, combined and sent to the outbound door on time for a kind of products, it will influence the normal transportation of other products.

### 3.2.1. Basic Assumptions:

- (1) Customers need products from at least one supplier and all customers' demands cover to each supplier. That is to say, each customer submits orders to one or more suppliers and each supplier supplies products to one or more customers.
- (2) When the products arrive in the cross docking distribution center, they are sorted and integrated immediately. The sorting and integrating time is thought of zero, because it's very short compared to transportation time. In conclusion, vehicles scheduling model is two-stage problem.
- (3) Direct shipment is adopted in two stages, namely, every vehicle only carries all products of one supplier or one customer.
- (4) All products of an order should be transported to the customer immediately once arrived in the distribution center.
- (5) The paper assumes that each customer has a truck in phase 2 and these trucks have nothing to do with phase 1, so there is no need to schedule the vehicles in phase 2.

### 3.2.2. Parameters:

$M$  : A number large enough

$i$  : Transportation stage

$n_i$  : Number of product in stage  $i$

$m_i$  : Number of vehicle in stage  $i$

$v_{ia_i}$  : Vehicle in stage  $i$ ,  $v_{ia_i} \in V_i = \{v_{i1}, \dots, v_{im_i}\}$ ,  $a_i = 1, \dots, m_i$

$j_{ie_i}$  : Product in stage  $i$ ,  $j_{ie_i} \in J_i = \{j_{i1}, \dots, j_{in_i}\}$ ,  $e_i = 1, \dots, n_i$

$p_{ie_i}$  : Transportation time of product  $j_{ie_i}$  in stage  $i$

$S_{e_2}$  : A set of precedent subset product of  $J_1$  corresponding to product  $j_{2e_2} \in J_2$ ,

$S_{e_2} \in S = \{S_1, S_2, \dots, S_{n_2}\}$

### 3.2.3. Decision Variables:

$C_{max}$  : Transportation time

$C_{ia_i e_i}$  : Transportation time of products  $j_{ie_i}$  in stage  $i$

$$x_{ia_i e_i} = \begin{cases} 1, & \text{if product } j_{ie_i} \text{ is assigned to vehicle } v_{ia_i} \in V_i \text{ in stage } i \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ie_i j_i} = \begin{cases} 1, & \text{if product } j_{ie_i} \text{ is transported before product } j_{if_i} \\ 0, & \text{otherwise} \end{cases}$$

### 3.2.4. Model II :

$$\min C_{max} \tag{9}$$

s.t.

$$\sum_{v_{ia_i} \in V_i} x_{ia_i e_i} = 1, i \in I, j_{ie_i} \in J_i \tag{10}$$

$$C_{ie_e} \geq p_{1e_1}, j_{1e_1} \in J_1 \tag{11}$$

$$C_{2e_2} \geq C_{1e_1} + p_{2e_2}, j_{2e_2} \in J_2, j_{1e_1} \in J_1, S_{e_2} \in S \tag{12}$$

$$C_{ie_i} + M (2 + y_{ie_i f_i} - x_{ia_i e_i} - x_{ia_i f_i}) \geq C_{if_i} + p_{ie_i}, i \in I, j_{ie_i}, j_{if_i} \in J_i, v_{ia_i} \in V_i \quad (13)$$

$$C_{if_i} + M (3 - y_{ie_i f_i} - x_{ia_i e_i} - x_{ia_i f_i}) \geq C_{if_i} + p_{if_i}, i \in I, j_{ie_i}, j_{if_i} \in J_i, v_{ia_i} \in V_i \quad (14)$$

$$C_{max} \geq C_{ie_i}, i \in I, j_{ie_i} \in J_i \quad (15)$$

$$x_{ia_i e_i} \in \{0, 1\}, i \in I, j_{ie_i} \in J_i, v_{ia_i} \in V_i \quad (16)$$

$$y_{ie_i f_i} \in \{0, 1\}, i \in I, j_{ie_i}, j_{if_i} \in J_i \quad (17)$$

The objective function (9) is to minimize the transportation time. Constraint (10) ensures that in every stage each product is assigned to exactly one truck. Constraint (11) indicates that each product is transported in the first stage. Constraint (12) ensures that each product  $j_{2e_2}$  in the second stage starts being transported only after the completion of any precedent subset product in  $S_{e_2}$ . Constraint (13) and (14) indicate direct shipment mode. A truck only carries a kind of products and there must be a sequence between two products. There is only one effective between constraint (13) and (14) when product  $j_{ie_i}$  and  $j_{if_i}$  are distributed to the same truck. Constraint (15) defines the maximum transportation time. Constraint (16) and (17) specify the domains of the decision variables.

### 3.3. Vehicles Scheduling Model with the Distribution Center to be Determined

In addition, this paper also considers another situation that the location of cross docking distribution center is determined based on the distribution of the main customers to deal with in future, namely, there is need to determine the location of the center first. The model of this situation is only need to introduce 0 – 1 variable  $x_i (i = 1, 2, 3, \dots)$  to the model in Section 3.2.

$$x_i = \begin{cases} 1, & \text{if supplier } J_{i1} \text{ is selected as distribution center} \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

## 4. Hybrid Particle Swarm Optimization

Chen and Lee [1] has proved cross docking vehicle scheduling problem is NP-hard which is generally solved by heuristic algorithm. Related research has been summarized in Section 1.

Particle swarm optimization (PSO) algorithm searches the optimal solution through the pursuit of individual extremum and group extremum. Particle swarm optimization algorithm has simple operation and fast convergence speed, but with the increase of the number of iterations, the population convergence is more and more concentrated, particle similarity is higher and higher, which will lead the particle to be in the local optimal solution and can't jump out. On the other hand, the genetic algorithm (GA) itself exists "premature convergence". So according to the advantages and disadvantages of the two algorithms, hybrid particle swarm optimization (HPSO) is constructed to solve the scheduling problem.

Hybrid particle swarm optimization divides the whole group into several sub groups, and each sub group is executed evolution operation as a whole. At the same time, it introduces clone selection operator to improve the performance of the algorithm on the basis of PSO and GA. Particles multiply and mutate by calculating the affinity between individuals so that the best individual can be reserved and the poor can be improved.

In order to better explain HPSO, detailed steps of the algorithm can be summarized as below:



- (1) Initialize the particle swarm. The population size is  $N$ , and the population is recorded as  $P_1$ .
- (2) Calculate the affinity of each antibody in  $P_1$  and sort descending order according to the affinity of antibody. Take half of the high affinity antibodies as the current group which noted for  $P_2$ .
- (3) For the individual in  $P_2$ , execute the cross operation of individual optimal and group optimal.
- (4) For the individual in  $P_2$ , execute the clone operation according to the values of affinity, and get group  $P_3$ . Clone quantity is proportional to the affinity of antibody.
- (5) With antibody reorganization operator to random recombine all antibodies in group  $P_3$ .
- (6) Mutation operation of group  $P_3$ .
- (7) Calculate the affinity of the mutated antibody and sorted by affinity. Take the first  $N$  of the high affinity antibodies which noted for  $P_4$ .
- (8) Update the current group. Merge  $P_2$  with  $P_4$  to get a new group  $P_1$ .
- (9) Repeat from steps (2)-(8) until iteration reaches their maximum limit or the stopping criterion is met.

Clone operator, crossover operator, antibody reorganization operator and mutation operator are designed in the algorithm.

#### 4.1. Clone Operator

Make particles correspond to the biological chromosomes, antibody correspond to the group containing  $n$  chromosomes, antigen correspond to the sub group with the optimal particle. Then the affinity between antibody and antigen is expressed as:

$$ay_i = \left( \sum_{j=i}^n F_j + l_{besti} \right) / 2 \quad (19)$$

Where  $F_j$  is the particle's fitness value in antibodies, and the fitness function is the objective function above mathematical models in section 3.  $l_{besti}$  is the biggest fitness value of particle in antibodies. Execute clone operation for high affinity antibodies by clone operator, and clone quantity  $N_c$  is given by:

$$N_c = \text{round}(\beta_n / i) + 1 \quad (20)$$

Where  $\beta$  is cloning coefficient which controls the clone size,  $\beta = 0.4 - 0.3g / g_{\max}$ . Where  $g_{\max}$  is the largest number of iterations,  $n$  is a constant, and  $i$  is the particle's order number according to the affinity descending order,  $g$  is the current number of iterations. It is seen from the type of expression (20) that clone size increases with the increase of affinity. The higher the affinity is, the more the clone number of antibody is. So the excellent individual with high affinity in group can survive and develop.

## 4.2. Crossover Operator

Crossover operation is an important way of genetic algorithm to obtain the offspring of outstanding individual. Individual updates itself through crossing individual and individual extremum, individual and group extremum. Integer crossing method is adopted here. Choose two cross positions first, and then cross individual and individual extremum or cross individual and group extremum. Assume that the random cross positions are 3 and 5. Operation method is as follows:

Individual: [9 4 2 1 3 7 6 1 0 8 5]

Group extremum: [9 2 1 6 3 7 4 1 0 8 5]

A new individual with Cross individual and group extremum: [9 4 1 6 3 7 6 1 0 8 5]

If there is a repeated position in the new individual, then adjust it. Adjustment method is to use the code not including in individual to take the place of repeated coding.

The individual above was adjusted to get a new individual: [9 4 2 1 3 7 6 1 0 8 5].

The strategy of keeping excellent individual is adopted here. Update particle only when the new particle's fitness value is better than the old particle's.

## 4.3. Antibody Reorganization Operator

Antibody reorganization operator randomly divides the whole group into several sub groups. It makes the particles in different sub groups into the same group. It makes the particles exchange the information among sub groups, which can expand searching space and search the global optimal solution. It is easy to fall into local optimal solution if particles neighborhood range remains the same.

## 4.4. Mutation Operator

Mutation is an important way to increase the population diversity. Moderate variation can not only maintain individual diversity of the population, but also improve the local search ability. Variation adopts exchange of two positions within the individual. Choose random mutation positions of position1 and position 2 first, and then exchange the two mutation positions. Assume that the mutation positions are 2 and 4. Mutation operation is as follows:

Individual: [9 4 2 1 3 7 6 1 0 8 5]

A new individual by mutating: [9 1 2 4 3 7 6 1 0 8 5]

The strategy of keeping excellent individual is adopted here. Update particle only when the new particle's fitness value is better than the old particle's.

## 5. Computational Experiments

In order to verify the effectiveness of the model and the feasibility of the algorithm, the paper uses the following computational experiments to analyze it. In the experiments, for programming convenience, the transportation time is transformed into the distance which can be computed based on the suppliers' and customers' locations. Therefore, a customer's transportation time is transformed into transportation distance, namely, a customer's transportation distance equals cumulative distance in the first stage starting delivering products for the customer and the distance from distribution center to customer. Fitness function value is correspondingly transformed into the minimum of customer's transportation distance.

Assume there are 21 suppliers and 10 customers whose locations are shown in Table 2 and Table 3.

**Table 2. Suppliers Location**

Number	1	2	3	4	5	6	7	8	9	10	11
X	58	51	52	31	5	12	36	52	27	17	13
Y	58	51	33	32	25	42	16	41	23	33	13
Number	12	13	14	15	16	17	18	19	20	21	
X	57	62	42	16	8	7	27	30	43	42	
Y	58	42	57	57	52	38	68	48	67	41	

**Table 3. Customers Location**

Number	1	2	3	4	5	6	7	8	9	10
X	45	59	5	10	21	5	30	39	32	25
Y	35	15	6	17	10	64	15	15	39	32

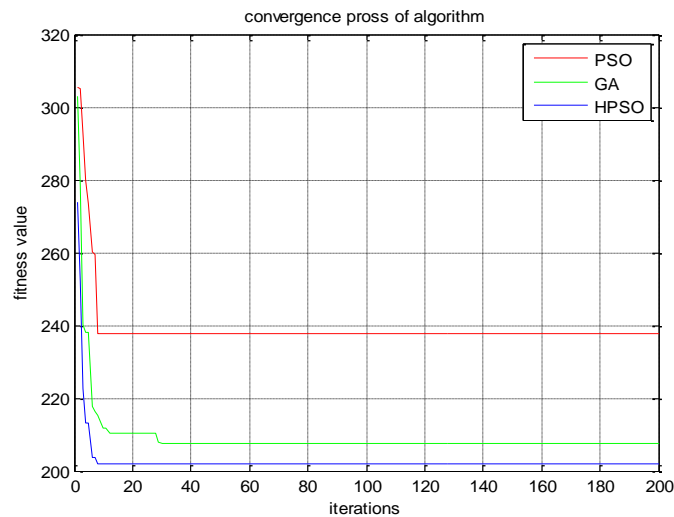
**5.1. The Result of a Single Vehicle Scheduling Model**

Set the origin as the distribution center. The set of customer’s order demand is generated randomly as Table 4.

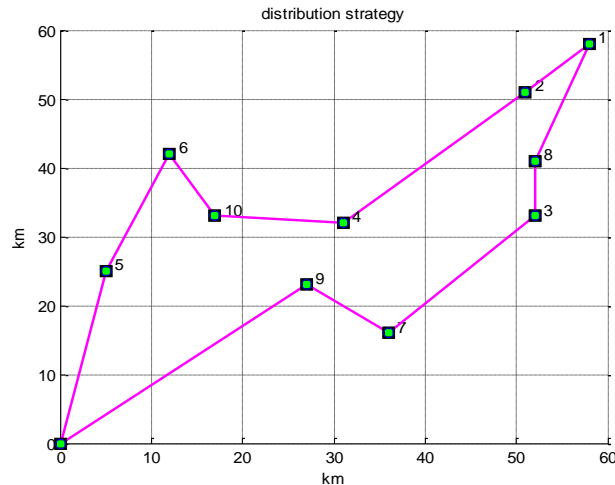
**Table 4. The Set of Customers Demand**

Customer	1	2	3	4	5	6	7	8	9	10
Supplier	9	6	5	10	2	8	3	7	1	4

For a single vehicle scheduling model, PSO, GA, HPSO are adopted in the paper to calculate it. The parameters of each algorithm are as follows: population is 50 and iteration number is 200 in PSO; population is 100 and iteration number is 200 in GA; population is 100 and iteration number is 200 in HPSO. Program to solve the model on MATLAB platform and the results are as follows: Figure 3 shows the convergence process of PSO, GA, HPSO; Figure 4 shows the distribution strategy obtained by HPSO; Table 5 shows the results comparison of three algorithms by running 30 times on this issue.



**Figure 3. Convergence Process of Algorithms**



**Figure 4. Distribution Strategy**

It is seen from Figure 3 that PSO has fast convergence speed, but is easy to fall into local optimum. The convergence speed of GA is slow and the effect of optimal solution is common; HPSO not only has faster convergence speed, and can get optimal solution of higher precision. Figure 4 shows the distribution strategy obtained by HPSO, namely the sequence of vehicles picking up products among suppliers is:5,6,10,4,2,1,8,3,7,9. And the optimal transportation distance is 201.9399km. Table 5 reflects HPSO has obvious superiority in convergence speed and the precision of optimal solution.

**Table 5. Algorithms Comparison**

Algorithm	Optimal solution	Average optimal solution	Iteration number obtained optimal solution
PSO	237.8779	251.0908	8
GA	201.9399	205.3237	21
HPSO	201.9399	201.9399	5

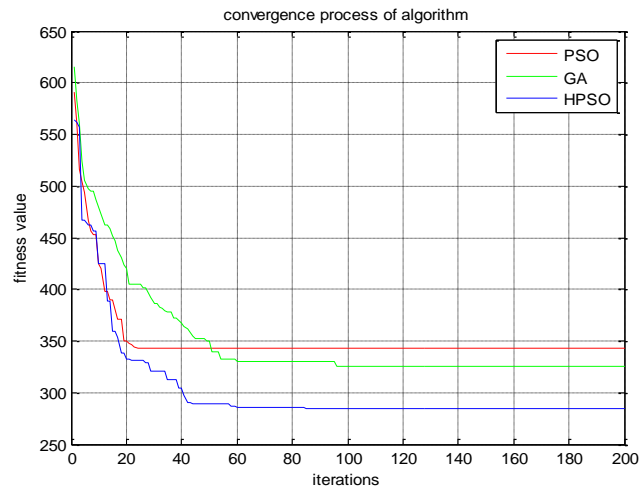
## 5.2. The Result of Vehicles Scheduling Model

Set the origin as the distribution center. The sets of customer's order demand are generated randomly as follows:

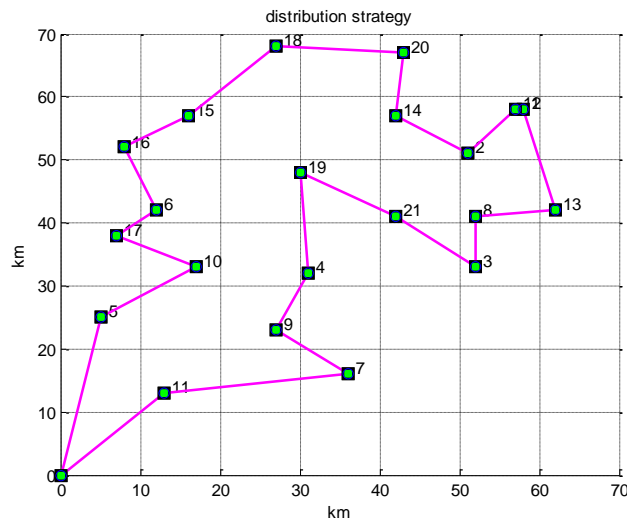
- Customer 1: {2, 4, 11, 1, 8}
  - Customer 2: {2, 7, 13, 15, 4, 16, 11, 9, 8, 6, 20, 3, 14, 5}
  - Customer 3: {21, 5, 10, 3, 14, 7, 1, 15}
  - Customer 4: {14, 5, 15, 7, 1, 4, 21}
  - Customer 5: {20, 10, 4, 9, 13, 11, 19, 8, 17, 5, 21, 1, 15, 7, 6, 16, 12}
  - Customer 6: {5, 12, 17, 2, 1, 4, 8, 7, 21, 3, 6, 19, 16, 9, 11, 18, 15}
  - Customer 7: {14, 6, 7, 2, 19, 18, 10, 15, 20, 11, 9, 3, 8, 4, 13}
  - Customer 8: {12, 3, 19, 2, 9, 17, 1, 5, 4, 16, 11, 21, 18, 10, 15}
  - Customer 9: {9}
  - Customer 10: {5, 9, 8, 2, 1, 3, 6, 7, 12, 17, 13, 11, 19, 14, 21, 4}
- Operation time is 259.381431s.

In a similar way, PSO, GA, HPSO are adopted to calculate the vehicles scheduling model. The parameters of each algorithm are as same as that in Section 5.1.

The results are as follows: Figure 5 the convergence process of PSO, GA, HPSO. Figure 6 shows the distribution strategy obtained by HPSO. Table 6 shows the results comparison of three algorithms by running 30 times on this issue.



**Figure 5. Convergence Process of Algorithms**



**Figure 6. Distribution Strategy**

**Table 6. Algorithms Comparison**

Algorithm	Optimal solution	Average optimal solution	Iteration number obtained optimal solution
PSO	342.7947	370.8293	28
GA	286.0809	325.6125	140
HPSO	280.4811	289.6054	48

It is seen from Figure 5 that PSO has fast convergence speed, but its optimal solution is unsatisfied. The convergence speed of GA is slow and it is easy to fall into the local optimal solution; HPSO not only has faster convergence speed, and can get the optimal solution of higher precision.

Figure 6 shows the distribution strategy obtained by HPSO, namely the sequence of vehicles picking up products among suppliers is:

5,10,17,6,16,15,18,20,14,2,12,1, 13,8,3,21,19,4,9,7, 11.

And the optimal transportation distance is 280.4811km. Table 6 reflects HPSO has obvious superiority in convergence speed and the precision of optimal solution.

### 5.3 The Result of Vehicles Scheduling Model with the Distribution Center to be Determined

The sets of customer's order demand are generated randomly as follows:

Customer 1: {12, 11, 18}

Customer 2: {6, 3, 4, 15, 5, 2, 8, 10, 9, 20, 18, 1}

Customer 3: {13, 9, 5, 3, 2, 18, 6, 16, 17, 10, 11, 14, 1, 4, 19, 8, 12, 21, 20, 15}

Customer4: {19, 1, 13, 20, 4, 17, 21}

Customer5 :{10, 2, 12, 13, 21, 8, 9, 7, 15, 19, 3, 6, 4, 18, 5, 17, 16, 1, 14, 20,11}

Customer 6: {17, 11, 9, 1, 14, 5, 21, 8, 19, 16, 18, 2, 13, 20, 10}

Customer 7: {7, 6, 17, 10, 1, 5, 15, 9, 11, 13, 21, 3, 14, 16, 18, 20, 12, 4, 8, 19}

Customer 8: {9, 4}

Customer 9: {11, 18, 21, 9, 15, 5, 6, 8, 14, 13, 20, 16, 7, 3, 10, 12, 4, 2}

Customer 10: {13, 11}

Assume that choose the near location of supplier 4 as distribution center because supplier 4 serves to more customers. Then in a similar way, PSO, GA, HPSO are adopted to calculate the vehicles scheduling model. The parameters of each algorithm are as same as that in Section 5.1.

The results are as follows: Figure 7 shows the convergence process of PSO, GA, HPSO. Figure 8 shows the distribution strategy obtained by HPSO. Table 7 shows the results comparison of three algorithms by running 30 times on this issue.

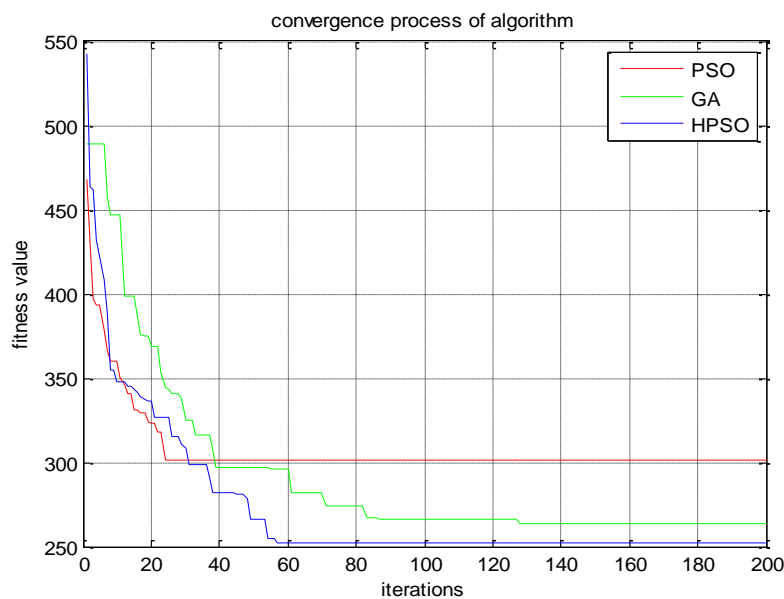
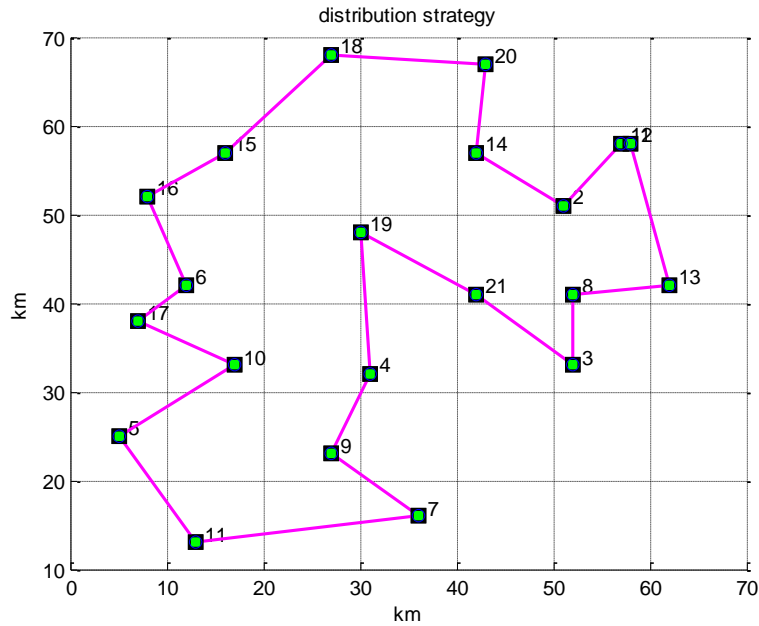


Figure 7. Convergence Process of Algorithms



**Figure 8. Distribution Strategy**

**Table 7. Algorithms Comparison**

Algorithm	Optimal solution	Average optimal solution	Iterations obtained optimal solution
PSO	301.4393	331.1439	24
GA	257.7820	284.7264	121
HPSO	251.0235	256.4589	43

It is seen from Figure 7 that PSO has fast convergence speed, but its optimal solution is unsatisfied. The convergence speed of GA is slow and the effect of the optimal solution is worse than HPSO's. HPSO not only has faster convergence speed, and can get the optimal solution of higher precision.

Figure 8 shows distribution strategy get by HPSO, namely the sequence of vehicles picking up products among suppliers is:

4, 9,7,11, 5,10,17,6,16,15,18,20,14,2,12,1,13,8,3,21,19

And the optimal transportation distance is 251.0235km. Table 7 reflects HPSO has obvious superiority in convergence speed and the precision of optimal solution.

## 6. Conclusion

This paper assumes that there are 21 suppliers and 10 customers. There will be 21!\*10! situations in which the calculated quantity is large and the calculated time is long if it adopts some enumeration methods. What's worse, calculated time and calculated quantity will increase quickly along with the increase of the number of vehicles. So it's difficult to get the optimal solution within the short time and limited iteration numbers. The results in section 5 indicate that hybrid particle swarm optimization algorithm not only has fast convergence speed, also avoids premature phenomenon well.

In the paper, we study the problem of two-stage cross docking scheduling under direct shipment mode. According to existing research, we first establish three mathematical models called a single vehicle scheduling, multiple vehicles scheduling and multiple vehicles scheduling of the distribution center to be determined with the objective to minimize transportation time. We then propose hybrid particle swarm optimization to

solve these models. Clone operator, crossover operator, antibody reorganization operator and mutation operator are introduced to improve the algorithm's performance. Computational experiments show that hybrid particle swarm optimization can optimally solve these problems. The improved algorithm is effective and feasible. It has faster convergence speed, and can get the optimal solution of higher precision. At the same time, it also avoids the premature phenomenon well. The models and the algorithm in this paper can provide a new way of solving the problem of cross docking scheduling.

## Acknowledgments

Our work is supported by the Fundamental Research Funds for the Central Universities (CDJZR12170014), the Chongqing basic and frontier research project (cstc2013jcyjA70006), and the National Natural Science Foundation of China (Grant No. 71201178).

## References

- [1] F. Chen and Chung-Yee Lee, "Minimizing the makespan in a two-machine cross-docking flow shop problem," *European Journal of Operational Research*, vol. 193, no. 1, (2009), pp. 59-72.
- [2] F. Chen and K. Song, "Minimizing makespan in two-stage hybrid cross docking scheduling problem," *Computers & Operations Research*, vol. 36, no. 6, (2009), pp. 2066-2073.
- [3] D.-Y. Ma and F. Chen, "Dynamic programming algorithm on two machines cross docking scheduling," *Journal of Shanghai Jiao Tong University*, vol. 41, no. 5, (2007), pp. 852-856.
- [4] W. Yu and P.-J. Egbelu, "Scheduling of inbound and outbound trucks in cross docking systems with temporary storage," *European Journal of Operational Research*, vol. 184, no. 1, (2008), pp. 377-396.
- [5] N. Boysen, M. Flidner and A. Scholl, "Scheduling inbound and outbound trucks at cross docking terminals," *OR Spectrum*, vol. 31, no. 1, (2008), pp. 135-161.
- [6] J. Van Belle, P. Valckenaers, G. V. Berghe and D. Cattrysse, "A tabu search approach to the truck scheduling problem with multiple docks and time windows," *Computers & Industrial Engineering*, vol. 66, no. 4, (2013), pp. 818-826.
- [7] M. Shakeri, M. Low, S.-J. Turner and E.-W. Lee, "A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained cross dock," *Computers & Operations Research*, vol. 39, no. 11, (2012), pp. 2564-2577.
- [8] N. Boysen, "Truck scheduling at zero-inventory cross docking terminals," *Computers & Operations Research*, vol. 37, no. 1, (2010), pp. 32-41.
- [9] R. Larbi, G. Alpan, P. Baptiste and B. Penz, "Scheduling cross docking operations under full, partial and no information on inbound arrivals," *Computers & Operations Research*, vol. 38, no. 6, (2011), pp. 889-900.
- [10] R. Musa, J.-P. Arnaout and H. Jung, "Ant colony optimization algorithm to solve for the transportation problem of cross-docking network", *Computers & Industrial Engineering*, vol. 59, no. 1, (2010), pp. 85-92.
- [11] A. Bellanger, S. Hanafi and C. Wilbaut, "Three-stage hybrid-flow shop model for cross-docking," *Computers & Operations Research*, vol. 40, no. 4, (2013), pp. 1109-1121.
- [12] S. M. Mousavi and R. Tavakkoli-Moghaddam, "A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain," *Journal of Manufacturing Systems*, vol. 32, no. 2, (2013), pp. 335-347.
- [13] A.-B. Arabani, M. Zandieh and S. Ghomi, "Multi-objective genetic-based algorithms for a cross-docking scheduling problem," *Applied Soft Computing*, vol. 11, no. 8, (2011), pp. 4954-4970.
- [14] G. Dong, J. Tang, K.-K. Lai and Y. Kong, "An exact algorithm for vehicle routing and scheduling problem of free pickup and delivery service in flight ticket sales companies based on set-partitioning model," *Journal of Intelligent Manufacturing*, vol. 22, no. 5, (2011), pp. 789-799.
- [15] F.-A. Santos, G.-R. Mateus and A.-S. da Cunha, "The pickup and delivery problem with cross-docking," *Computers & Operations Research*, vol. 40, no. 4, (2013), pp. 1085-1093.
- [16] B. Vahdani and M. Zandieh, "Scheduling trucks in cross-docking systems: Robust meta-heuristics," *Computers & Industrial Engineering*, vol. 58, no. 1, (2010), pp. 12-24.
- [17] A.-R. Arabani, S. Ghomi and M. Zandieh, "Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage," *Expert Systems With Applications*, vol. 38, no. 3, (2011), pp. 1964-1979.
- [18] T. -W. Liao, P.-J. Egbelu and P.-C. Chang, "Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations," *Applied Soft Computing*, vol. 12, no. 11, (2012), pp. 3683-3697.

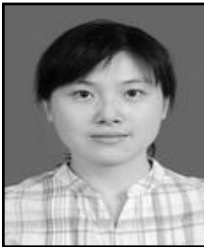


- [19] T. -W. Liao, P.-J. Egbelu and P.-C. Chang, "Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations," *International Journal Of Production Economics*, vol. 141, no. 1, (2013), pp. 212-229.
- [20] B. Vahdani, R. Tavakkoli-Moghaddam, M. Zandieh and J. Razmi, "Vehicle routing scheduling using an enhanced hybrid optimization approach," *Journal of Intelligent Manufacturing*, vol. 23, no. 3, (2012), pp. 759-774.
- [21] C.-H. Wang and J.-Z. Lu, "An effective evolutionary algorithm for the practical capacitated vehicle routing problems," *Journal of Intelligent Manufacturing*, vol. 21, no. 4, (2010), pp. 363-375.

## Authors



**Hairu Zhao**, born in 1989, is a master student from Automation College in Chongqing University. Her major is control engineering. Her primary research interests include logistics system, information system and intelligent optimization algorithm.



**Ling Chen** received her Ph.D. degree from Chongqing University in management science and engineering. She is a Lecture at Logistics Engineering department of Automation College in Chongqing University. Her research interests include logistics operation, operational research, optimization method, and automation technique in the field of logistics procedure optimization.

