Dynamic slotting optimization based on SKUs correlations in a zone-based wave-picking system

LI Yingde[a]

[a]Institute of industrial engineering, Zhejiang University of technology, Hangzhou, Zhejiang, P.R. china, 310014

Jeffery S. Smith[b,*]

[b]Department of industrial & system engineering, Auburn University, Auburn , Alabama,usa,36849-5386

Abstract

Many exiting slotting methods ignore the picking correlations between Stock Keeping Units (SKUs). In a previous paper, a mix integer program model for dynamic slotting to minimize the pick-wave makespan among all zones under some load balancing constraints was developed. In this paper, we develop an ant colony optimization with slot-exchange policy (ACO-SE) based on SKU correlation to assign the correlated SKUs to the adjacent slots in the same zone. The ACO-SE deposits pheromones between SKUs, uses local and global pheromone trail updates, and controls pheromone accumulation using the Max-Min rule. The main heuristic information is set to the correlation strength and the pick-times are introduced as the assisted heuristic information. A hybrid search mechanism was adopted to improve to global search efficiency. A slot exchange policy was proposed to re-slot the correlated SKUs based on the picks to ignore the proximity of SKUs and to make the farthest SKU for one carton closer to the initial point as far as possible. The promising computational results show that the ACO-SE has perfect convergence and very good CPU time. The solution quality of ACO-SE is always better than the Cube-per-Order-Index (COI), simulated annealing correlation (SA-C) heuristic; it has considerably faster convergence speed than SA-C. The result shows that in zone-based wave-picking system with return touring policy, the exact proximity of SKUs is not critical and that the correlated SKUs can be allocated to any locations along the path from the initial point to the other SKU's location; the correlation strength has no obvious impact on the picking efficiency, but and correlation probability

[*] Corresponding author - Tel.:+1 334 844-1412
E-mail: liyingde2008@sina.com (Li Yingde), jsmith@auburn.edu (Jeffery S. Smith)

has significant impact on the picking efficiency.
Key Words: Dynamic slotting; SKUs correlation; ACO-SE; Correlation strength; Correlation probability; Proximity

1 Introduction

A survey by Frazelle [7] has demonstrated that order picking accounts for over 50% of the total operating cost in a typical warehouse. De Koster *et al* [3] give a comprehensive literature review on this topic. The warehouse environment on which we focus is a zone-based, picking system where cartons travel from zone to zone and pickers in each zone pick items into the cartons ([16]). Figure 1 illustrates this system. When a carton arrives at a zone, the picker takes a carton from the *zone initiation point*, scans the carton bar code so that the warehouse management system (WMS) can identify the SKUs to be picked in that zone and light the corresponding pick-to-light lights in the picking area. The picking policy is the Return travel policy – the picker walks down the aisle picking the required SKUs, places the carton on the conveyor to be transported to the next zone, and then returns to the initiation point and repeats the process for the next carton.

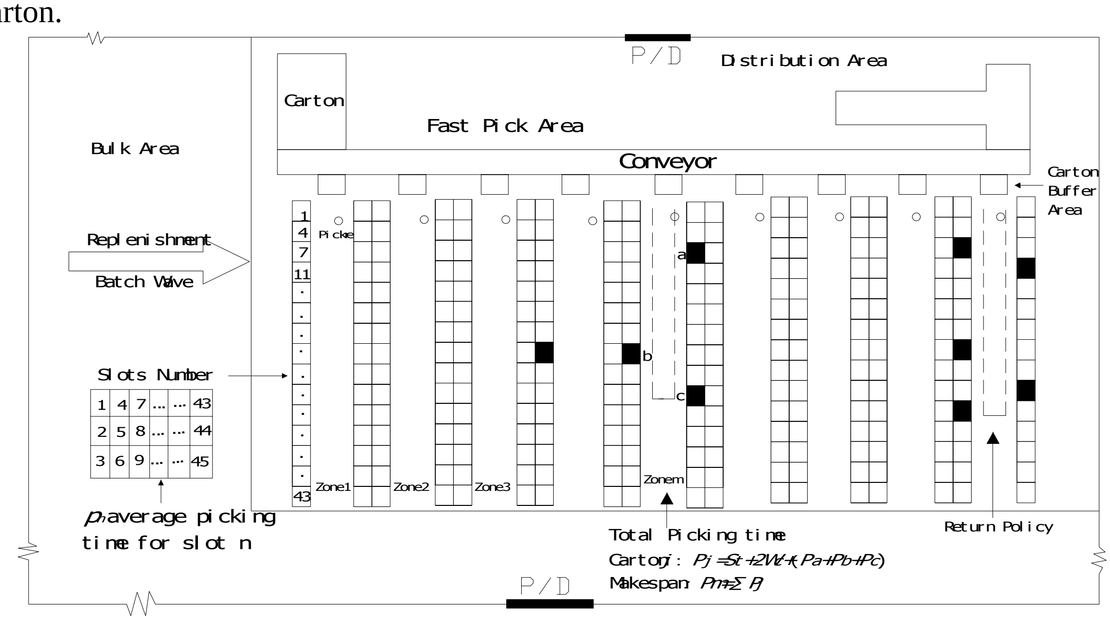


Figure 1. Configuration of dynamic pick-wave zone-based warehouse.

One interesting aspect of this system is that the picking area is completely setup for and emptied during each pickwave. That is, the picking area is relatively small and different sets of items are picked on different days and, between picking shifts the picking area is completely replenished specifically for the subsequent pickwave. The *slotting problem* involves determining an assignment of SKUs to slots in the picking area. Since the picking area is completely replenished for each pickwave, the slotting problem must be explicitly solved between each pickwave. As described in Kim and Smith [16], we are interested in developing a slotting methodology that minimizes the pickwave makespan in this dynamic environment.

Customer orders are divided into cartons, where each carton typically contains multiple SKUs in that order. Depending on the slotting, cartons may travel to multiple zones. A pickwave consists of the set of cartons comprising the orders for that day (referred to as the "carton list"). The pickwave makespan is the time required to pick all cartons in the carton list for the pickwave. As cartons move through the system, they incur the following "costs" (time):

1. Travel time between zones;
2. Zone initiation time – the time for the picker to pick up the carton, scan the barcode, and wait for the WMS to identify the order and light the picking lights; and
3. Pick time for SKUs – for each SKU, this includes the time for the picker to walk from his/her current location to the specified slot and pick the quantity of items into the carton.

As in Kim and Smith [16], we ignore the first component assuming that, with a large number of cartons in the system, each zone will have a large queue of cartons waiting such that the carton travel time does not substantially contribute to the makespan. With this assumption, we can compute the picking time for each zone (the sums of the initiation times and picking times for all cartons that visit that zone) and the makespan will be the largest zone picking time. As such, the slotting problem in which we're interested, involves assigning the SKUs to slots so that the largest zone picking time is minimized.

The premise of our work (as with Kim and Smith [16]) is that we can exploit SKU correlations – i.e., cases where multiple SKUs are commonly picked to the same cartons by assigning these SKUs to the same zone to minimize the number of zone initiations and the walk time associated with picking. However, where Kim and Smith (2012) assumed that SKU adjacency within a zone is important, we generalize this to show that multiple slots within the zone are equally good in terms of improving the makespan. Further, the ant colony optimization (ACO) approach performs better than previous approaches in both solution quality and solution time/scalability.

The remainder of the paper is organized as follows: Section 2 briefly discusses the related literature and focus on the optimization formulation from Kim and Smith [16]; In Section 3, we propose an ACO with a slot exchange policy to solve the dynamic slotting problem based on SKU's correlation; the experimental results are reported in Section 4; finally, Section 5, presents conclusions and further research ideas.

2 Background

About the slotting (storage assignment) problem, much research has been done and several papers have been written – i.e., see Petersen [22], [23], Heskett [12], [13], Harmatuck [10], Malmborg [19], [20] and Hwang *et al.* [14]. However, much the previous research ignores the SKU correlations and focuses on cube-per-order index (COI)-based methods. But in picking systems where multiple items are picked to the same carton, there are potential time savings by exploiting the SKU correlations during slotting.

In a *static demand* system, the incoming and outgoing of SKU flow patterns are relatively stationary over the planning horizon. Frazelle and Sharp [4], [5], developed a procedure to assign SKUs to slots based on the correlations among SKUs in this environment. Their research focused on developing a statistical correlation measure to use forming clustering of SKUs and the results shown that the SKUs that are likely to appear in the same order should be stored in nearby slots. Malmborg [20] developed the slotting with zone constraints and a heuristic procedure for using the COI to generate an initial item assignment followed by an improvement step using the Simulated Annealing (SA) algorithm. Zhang and Bo [28] discussed how to find a right place for SKUs firstly when we export, import goods or change a site of goods under the practical experiences in an automated three-dimensional warehouse. Xiao and Zheng [27] considered both material relevancy and requirement frequency, proposed a Hybrid Genetic Algorithm (HGA) to solve the static slotting problem in multi-aisle picking system. Liu *et al.* [18] and Bie and Li [2] also discussed the slotting problem in the Automatic Storage Retrieval System (AS/RS). Most of existing research studied the storage policy for Automated Storage and Retrieval system or picker-to-part system while few considered the pick-and-pass system.

In a *dynamic demand* system, the patterns of SKUs flow changes dynamically or periodically due to the factors such as seasonality, life-cycle or turnover rate, the slotting location of SKUs should be adjusted to reflect the changing in time. In the dynamic environment, once the order and cartonization information have been given, the number of SKUs and correlations between them can be exploited during slotting. Hackman and Platzman [11], Frazelle *et al*. [6], Van den berg *et al*. [26], and Bartholdi and Hackman [1] studied the fast pick replenishment planning problem from the reserved (Bulk) area. The main decisions were to select how much and where of each SKU should be stored in the restricted small fast pick area. In order to exploit the difference between products in terms of inventory profiles and usage patterns, Goetschalckx and Ratliff [9] developed a shared slotting policy for a unit load warehouse where over time different SKUs are stored in the same slot. Under the less than unit load picking, Landers *et al.* [17] and Sadiq *et al*. [24] considered a dynamic system where products evolve through a life cycle and thus the products mix varies over time, which creates a need to resize SKU slots and re-slotting. They proposed a procedure that included a clustering algorithm to decide which SKU should be stored together based on the long-run average correlation. But in the dynamic picking system, the information provided by long-run average demands may potentially lead to inefficient slotting.

Literature on specific dynamic slotting is not abundant. As mentioned previously, Kim and Smith [16] proposed an efficient slotting mythology under whole warehouse dynamic replenishment system. Using the correlations between SKUs, they proposed a

MIP formulation, whose objective is to minimize the pick wave makespan–the maximum total completion time among all pickers. As the problem is NP-hard, they developed a correlated slotting improvement heuristic (called SA-C) based on simulated annealing. The SA-C can potentially avoid the local optima and the analysis results shown the SA-C can achieve promising improvements. However, for medium and large problems, SA-C is computationally expensive since in only considers swaps of size two.

Further, in the SA-C, correlated SKUs that have strong correlations are assigned to *adjacent* slots. However, in a zone-based system with the *Return* travel policy, the "proximity" of the SKUs to one another is not really important in reducing pick time. Instead, the SKUs just need to be assigned to the same zone and one of the correlated SKUs can be allocated to *any* location along the path from the zone initiation point to the other SKU's location. So, when considering SKU exchanges, we have many more potential slots to consider in order to improve the overall solution.

3 ACO with slot-exchange policy based on SKUs correlations

We use the same mix-integer program formulation proposed by Kim and Smith (2012) to improve the picking efficiency in the same picking system and we present an improved ant colony optimization (ACO). The ACO uses the proper ant tour diagram, heuristic information, and a hybrid search mechanism to construct the feasible solution that is a sequence of SKUs based on their correlations. Next, we will propose a slots-exchange policy that ignores the specific "proximity" of the SKUs in the same zone to improve the picking efficiency, and to compare the results with SA-C heuristic from Kim and Smith (2012), and to perform some analysis on the impacts of the correlation on the picking efficiency. In the next section, an improved ant colony optimization based on SKUs correlations will be proposed. We call this procedure ACO-SE.

The standard ACO abstracts the problem as a node diagram through which artificial ants make tours. Each completed tour is a feasible solution of the problem and, with the feedback of pheromones the solutions will gradually convergence the optimal solution. Based on the dynamic slotting problem, we will make some modifications to improve the performance of the standard ACO. The objective of ACO-SE is to find the slotting with minimum makespan.

3.1 Construction of diagram and pheromone

As shown in Fig.2, the slotting problem can be represented as a complete linked diagram G=(A, L), A is the SKUs set, L is arcs set of two adjacent SKUs. The procedure of assigning SKUs to slots can be seen as the ant moving in the diagram guided by the constraints, the pheromone trail and the heuristic information. A completed tour will generate a SKU sequence in which SKUs are assigned to slots, thus the tour represents a feasible solution (i.e. the right part of Figure 2) and the makespan of the pick wave can be computed for a given number of orders.

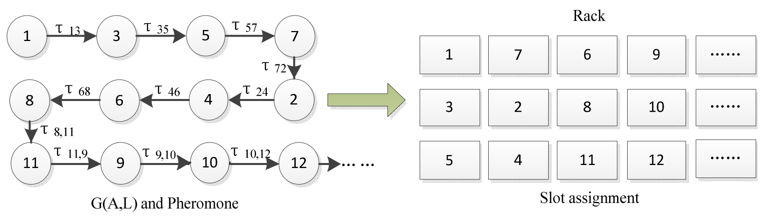


Figure 2. Diagram, pheromone and slot assignment

The ants deposit pheromones between two adjacent SKUs. The pheromone τ_{ij} shown in Figure 2, is set based on the proportion that SKU *i* and *j* are assigned to the adjacent slots in the same zone. A higher value indicates that there were more previous ants assigning SKUs *i* and *j* to adjacent slots. As this value increases so does the probability that current and subsequent ants assign the SKUs to adjacent slots and increase the pheromone on this route even more.

3.2 Heuristic information

In order to reduce the computational time, we will use some special information based on the slotting problem which will help the ants to construct the feasible solution rapidly and speed up the solution convergence. In accordance with the characteristics of the slotting problem based on SKUs correlation, we chose the *correlation strength* as the main heuristic information and the *picks* of SKU as the assisted heuristic information.

Correlation weight $C(i,j)$ represents the average correlation between SKUs *i* and *j* (Kim and Smith, 2012). These weights are used to generate random problem data – the greater the $C(i,j)$ is, the stronger the correlation between SKU *i* and *j*. The correlation probability F_i is defined as the proportion of correlated SKUs with SKU *i* to the total SKUs. The greater the F_i is, the more correlated SKUs there are with SKU *i*. The correlation strength $C(i,j)$ shows that how strong is the correlation between SKU *i* and *j*, the correlation probability F_i shows that how many SKUs are correlated with SKU *i*. Thus, F_i and $C(i,j)$ decide the correlation among all SKUs in a given pick wave. The correlation strength is decided by K (the total number of SKUs), *g* (the average line-items per carton), J (the total number of cartons) and F_i. For a set of given orders, F_i can be explored by the order information, the smaller the *g* is, the greater the *J* and $C(i,j)$ are.

We define the picks P^i_{sum} as the total number of cartons that contain SKU *i* (i.e., the total number of picks of SKU *i*). Clearly, the more popular SKUs have larger P^i_{sum} and should be assigned to the more convenient locations. In ACO-SE, P^i_{sum} is the assisted heuristic information used to develop the solution.

3.3 Construction of a feasible solution

In a pick wave, for a given SKUs set $\mathbf{A}=\{i \mid i=1,2,3......,K\}$ and the correlation information exploited based on the carton packing list, we design a procedure for constructing a feasible solution which is a SKUs sequence based on an ant tour in the diagram G. The procedure is as follows:

Step (1): Initialize the pheromone $\tau_{ij}=\tau_0, i, j \in K$, τ_0 is the initiation pheromone; let zone 1 as the current zone, set the zone index *m*=1; let slot 1 as the current slot, set the slot index *n*=1;

Step (2): For the current zone *m*, ant randomly selects an unassigned SKU form the set **A** as the first SKU of zone *m*, let the selected SKU as the current SKU , mark it as SKU *i*; set *n*=*n*+1;

Step (3): Using the correlation information, select all unassigned SKUs which are correlated with current SKU *i* to form the candidate set $\mathbf{A0}=\{j \mid C(i,j)>0, i, j \in \mathbf{A}\}$; the remaining unassigned SKUs form the set $\mathbf{A1}=\{j \mid C(i,j)=0, i, j \in \mathbf{A}\}$;

Step (4): If $\mathbf{A0}=\varnothing, \mathbf{A1}\neq\varnothing$, which means that there are no correlated unassigned SKUs with SKU *i*, thus select randomly any unassigned SKU *j* form **A1** and assign it to the adjacent slot of current SKU, set SKU *j* as to the current SKU and *n*=*n*+1; If *n*=*N*, the current zone *m* has no unassigned slots, set *m*=*m*+1, go to step (2); otherwise go to step (3);

If $\mathbf{A0}\neq\varnothing$, which means there are one or more correlated unassigned SKUs with SKU *i*, the ant selects SKU *j* as the next SKU of the sequence based on the following hybrid search mechanism which is shown in equation (1):

$$j=\begin{cases} \mathrm{O1}: \arg\max\limits_{j\in\mathbf{A0}}([\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}), 0\le r\le r_1 \\ \mathrm{O2}: P_{ij}=\dfrac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum\limits_{j\in\mathbf{A0}}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}, r_1<r\le r_2 \\ \mathrm{O3}: \text{Random } j,\ \ j\in\mathbf{A0},\ \ r_2<r\le 1 \end{cases} \tag{1}$$

In equation (1), r~U($0,1$)is a random variable, r_1 and r_2 are user-defined control parameters, $0\le r_1\le r_2\le 1$; α and β are pheromone factor and heuristic information factor respectively; τ_{ij} is the pheromone between SKU *i* and *j*; the correlation strength is the heuristic information, set $\eta_{ij}=C(i,j)$.

Set the selected SKU *j* as the current SKU, set *n*=*n*+1; If *n*=*N*, it means that the current zone *m* has no unassigned slots, set *m*=*m*+1, go to step(2); otherwise go to step(3);

If $\mathbf{A0}=\varnothing, \mathbf{A1}=\varnothing$, which means that all SKUs have been assigned and the ant has completed a tour, a feasible solution is constructed; go to step (5);

Step (5): The ant has constructed a feasible solution, STOP the tour. The feasible solution is a SKUs sequence which consists of all SKUs in a pick wave. The sequence means all the SKUs are assigned to slots by some rules.

During the construction of feasible solution, the ant makes the best use of the correlation information to reduce the number of the candidate SKUs; the ant selects the next SKU only from the candidate set which has a relatively small number of correlated SKUs. In the TSP, the candidate set is all unvisited cities, so there is a very poor search speed when there are a large number of cities. In the ACO-SE, we introduce the correlation information among SKUs to achieve a high search speed by reducing the number of the candidate SKUs.

In order to alleviate the local traps and enhance the search capacity, we need to enlarge the search space and make best use of the known information, adjusting the main search direction onto the solution space where the optimal solution may be. Thus, a hybrid search mechanism is proposed in equation (1), the ant will execute the Max, Probability and Random search policy by the probability of r_1, r_1- r_2 and 1-r_2 respectively, these improved operations can enlarge the search space and capacity effectively.

3.4 Pheromone update rule

The pheromone local update reflects the reverse feedback, during the construction of feasible solution, the ant moves at each step which means that SKU *j* is assigned to the adjacent position of SKU *i*, the pheromone τ_{ij} between SKU *i* and *j* will be evaporated at some rate to reduce the impact on to subsequent ants and enhance the search capacity. The local update rule is shown as follows:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\tau_0 \tag{2}$$

In equation (2), ρ is the evaporate factor, and $0 \le \rho \le 1$; τ_0 is the initiation pheromone.

The pheromone global update reflects the positive feedback, which means an extra reward to the ant who finds the current best solution to encourage more subsequent ants to select the same assignment with higher probability. Different from the standard ACO, the ACO-SE allows only the *current global ant* to deposit pheromone to alleviate the premature and overcome the local-best trap, which means, in each iteration, we uprate the pheromone with equation (3) for the current global-best solution.

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}^{gb} \tag{3}$$

In equation (3), if the arc (*i*,*j*) is included in the current global-best solution, set $\Delta\tau_{ik}^{gb} = \dfrac{Q}{T_m^{best}}$, the T_m^{best} means that objection value of the current global-best solution (i.e. best makespan), Q means the pheromone strength, it is a user-defined control parameter; otherwise, set $\Delta\tau_{ik}^{gb} = 0$.

3.5 Steps of the improved ACO with slot- exchange policy

The general steps of the ACO-SE based on the correlation are as follows:

Step (1): Initialize the parameters, including $r_1, r_2, \rho, M_{aco}, \tau_0, \tau_{ij}, It_m, \alpha, \beta, \tau_{\max}, \tau_{\min}, Q$. M_{aco} is the total number of artificial ants, It_m is the maximum iteration times; $\tau_{\max}$ and $\tau_{\min}$ are the maximum and minimum pheromone;

Step (2): For each ant, construct a feasible solution in accordance with the method proposed in section 3.3; during the construction, when the ant moves at each step, we need to update the local pheromone trail with the equation (2). In order to speed up the convergence and avoid the stagnation, we use the Max-Min rule to control the total pheromone quality: if $\tau_{ij} < \tau_{\min}$, then set $\tau_{ij} = \tau_{\min}$; if $\tau_{ij} > \tau_{\max}$, then set $\tau_{ij} = \tau_{\max}$;

Step (3): Assign all SKUs in the sequence (a feasible solution) to the slots in the zone one by one to generate an origination slotting assignment. So, the two SKUs which have stronger correlations are assigned to the adjacent slots, they are neighbors on the rack and the proximity is very small.

But in the zone-based system with *Return* travel policy, the “proximity” of the SKUs in the same zone is not important: they just need to be assign to the same zone. One of correlated SKUs can be allocated to any location along the path from the initial point to the other SKU’s location. A slot-exchange policy is needed to ignore the “proximity” in order to achieve a better picking efficiency.

Suppose the average picking time p_n is greater than the average time *v* of walking through one column, the origination slotting assignment needs to be improved by the following way: For any two adjacent slots N_1 and N_2, if slot N1 has more convenience than slot N2, and $P_{sum}^{N_1} < P_{sum}^{N_2}$, then we exchange the two SKUs in slot N_1 and N_2. The exchange operation does change the sequence constructed by ant touring in the diagram.

We call this procedure the *slot-exchange* policy and its purpose is to assign the SKUs that have more *picks* to the slots that are closer to the zone initiation point. After the exchange operation, a new improved slotting assignment has been constructed.

Use the slot-exchange policy to re-slot;

Step (4): For the new improved slotting assignment, calculate the objection value (i.e. makespan); set the ant who find the current best solution as the current best ant, if the gap among solutions is not obvious, set the ant who find the smallest standard deviation of completion time as the current best ant. Update the global pheromone with equation (3) and the Max-Min rule to control the total quality of pheromone;

Step (5): If the iteration times is more than It_{m}, STOP and output the result; otherwise, go to Step (2).

4 Test and analysis

4.1 Parameters setting and test methods

The ACO-SE is coded in Visual Basic 6 (SP6), the test operating system is Windows 7 64 bit, 8 GB of RAM, and and Intel Core^{TM} i5-520 (2.4 Ghz) CPU. We use the same picking system parameters as Kim and Smith (2012) to do our testing (see Table 1).

Table 1 Picking system parameters

Parameter/unit	Value	Parameter/unit	Value
Number of SKUs (K)	540/1080	Correlation strength ($C(i,j)$)	1,2,30
Number of zones (M)	10/20	Zone setup time (Zs) /s	43
Rack Levels (R_l) and columns (R_c)	3,9	Setup time for carton (S_t) /s	10.8
Average correlation probability (F_i)	0.05~0.15	Unit walking speed (v) /s/column	1.4
Number of slots (N)	54/108	Average line-items per carton (g)	2~20
Number of cartons (J)	100~700	Unit picking time(bottom/middle/top Level of rack) (p_n) /s	3.48/2.9/3.05

With many possible test combinations, we set the basic ACO-SE parameters as follows: $M_{aco} = M$, $\tau_{\max} = 10$, $\tau_{\min} = 0.01$, $\tau_0 = 0.01$, $Q = 1200$, $It_m = 500$. The other parameters such as $r_1, r_2, \rho, \alpha, \beta$ will affect the results significantly, it is important to do some analysis and combination test to decide the proper parameters.

When r_1 is larger, the convergence rate is better, but it is easier to fall into the local optimums. When r_1 is smaller, the searching speed is slower and it is easier to fall into the stagnation. When r_2 is smaller, ACO-SE cannot make use of the SKU correlation information and the convergence rate and the robustness of solution will be reduced. Table 2 shows the results for several combinations of the two search control parameters for a random problem (J=250, K=540, M=10, N=54, C(i,j)=2, F_i=0.10). In the “Evaluation” column, “Local optimum” means that ACO-SE fall into the local optimum trap, “Slow” means the convergence rate of ACO-SE is slow, “Near-best” means the ACO-SE solution is close to the best solution found. It is clear, when $r_1 = 0.5$, $r_2 = 0.8 \sim 0.9$, the ACO-SE can get better solution.

When the evaporate factor ρ is bigger and near to 1 or smaller and near to 0, the pheromone will be enhanced to fall into local optimum or weakened to fall into stagnation. Table 3 shows the test result of evaporate factor for a random problem (J=250, K=540, M=10, N=54, C(i,j)=2, F_i=0.1), when $\rho = 0.2 \sim 0.4$, the ACO-SE can find better solutions.

The pheromone factor α and heuristic factor β show the impact of pheromone traces and correlation strength on the ants’ decisions in selecting the nodes in their paths (i.e., SKUs). When α is larger, the ants select SKUs that were selected by previous ants with higher probability; when β is larger, the ants are more likely to select the SKUs that have strong correlations. Table 4 shows the combination tests results for a random problem (J=250, K=540, M=10, N=54, C(*i*,*j*)=2, F_i=0.1). It is clear that when $\alpha = 3, \beta = 1$, the ACO-SE finds the best solutions.

Table 2 Search control parameters setting (part)

r_1	r_2	Makespan	Iteration	Evaluation	r_1	r_2	Makespan	Iteration	Evaluation
0.10	0.30	2100.91	97	Local optimum	0.40	0.90	1888.60	202	Slow, Near-best
0.10	0.50	1996.92	145	Slow	0.50	0.80	1897.52	149	Near-best
0.30	0.70	1926.53	302	Slow	***0.50***	***0.90***	***1865.31***	***91***	***Best Found***
0.30	0.90	1884.81	208	Slow	0.60	0.80	1881.98	116	Local optimum
0.40	0.70	1930.64	44	Local optimum	0.60	0.90	1877.72	92	Local optimum

Table 3 Evaporate factor setting (part)

ρ	Makespan	Iteration	Evaluation	ρ	Makespan	Iteration	Evaluation
0.10	1892.40	51	Local-optimum	0.60	1887.67	276	Slow, Local-optimum
0.20	***1856.45***	***68***	***Best Found***	0.70	1865.39	306	Slow, Near-best
0.30	1860.07	56	Near-best	0.80	1891.30	163	Local-optimum
0.40	1865.62	73	Near-best	0.90	1890.61	183	Local-optimum

Table 4 Search control parameters setting (part)

α	β	Makespan	Iteration	Evaluation	α	β	Makespan	Iteration	Evaluation
1	5	1926.31	306	Slow	3	3	1854.81	31	Local-optimum
2	5	1837.92	269	Near-best, slow	4	2	1874.27	30	Local-optimum
2	4	1868.11	74	Local-optimum	4	3	1895.23	121	Local-optimum
3	4	1837.74	127	Near-optimum	***3***	***1***	***1823.21***	***96***	***Best Found***
1	3	1868.80	377	Slow	3	2	1933.89	23	Local-optimum
2	*3*	1883.02	188	Local-optimum	2	1	1832.13	110	Near-best

So, we use the other ACO-SE parameters as follows: $r_1 = 0.5$, $r_2 = 0.9$, $\rho = 0.2$, $\alpha = 3, \beta = 1$. We use three methods to evaluate the performance of the ACO-SE and the correlations impacts on the picking efficiency.

(1) *Positive test*: For a given group of carton lists and average line-items per carton (*g*), we can exploit J and K, and to calculate the makespan using the ACO-SE. In this situation, the correlation strength *C*(*i*,*j*) and correlation probability F_i can be exploited from the carton lists. The positive test is used to test and evaluate the performance of ACO-SE including the convergence, CPU time, robustness and efficiency.

(2) *Opposite test*: With the method proposed by Kim and Smith (2012), a group of

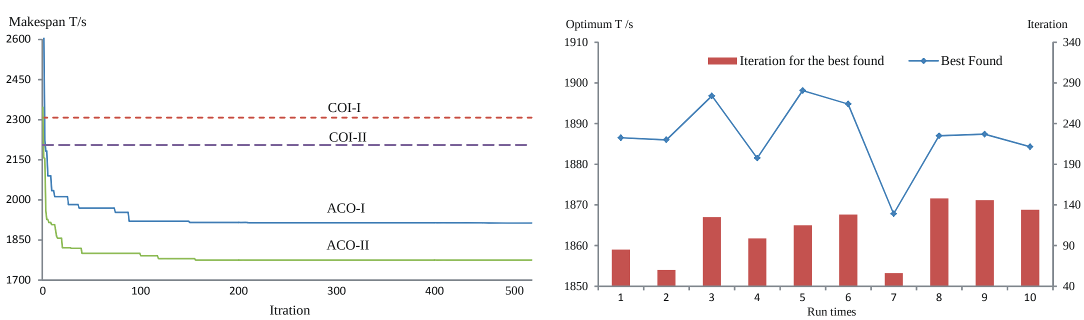
random carton lists is generated by controlling the five parameters which include J,K,$C(i,j)$,g and F_i. In this situation, we can exploit the correlation relations by analyzing these parameters. We will use this method to generate the testing data to evaluate the impacts of different correlations on the picking efficiency.

(3) *Comparison test*: we will compare the performance of three methods including the ACO-SE, COI and SA-C heuristic. The steps of COI storage policy based on the picks are as follows: first, calculate the P^i_{sum} for each SKU in a pick wave; second, sort the SKUs by P^i_{sum} in descending order; third, sort all slots by pick time (including walk time) in ascending order; fourth, sequentially assign the ordered SKUs to the ordered slots. Since the COI storage policy ignores the SKUs correlations, the results can be used to evaluate the impact of the correlations on the picking efficiency.

We use a number of randomly generated problem based on the experimental factors with several levels to evaluate the ACO-SE, since the complexity of the problem depends on K, J, C(i,j), g, F_i, we control these five parameters to several levels. The levels of each parameter are already shown in Table 1.

4.2 Performance test for small problems

We first use the positive and comparison tests to evaluate the performance of the ACO-SE. For a given group of orders – each order includes 2 to 15 line-items – we generate two groups of carton lists with g=5 and g=10 respectively, the first group (g=5, i.e. Scenario-I) consists of 216 small volume cartons and the second group (g=10, i.e. Scenario-II) consists of 118 big volume cartons. Thus, for the given orders, the two groups of carton lists have different correlations. Comparatively speaking, the first group has bigger $C(i,j)$ and smaller F_i; the second group has smaller $C(i,j)$ and bigger F_i, but the $C(i,j)$ and F_i are unknown. We ran the ACO-SE for the two scenarios respectively, Figure 3(a) shows the evolution curve for the two scenarios at the initial run-time, and Figure 3(b) shows the best found values and the iteration times (Histogram) of finding the optimum at each run time in the Scenario-I.



(a) Evolution curve for two scenarios at initial run-time (b) Stability of solution for the Scenario-I

Figure 3. Performance test results (Scenario-I: g=5, J=216; Scenario-II: g=10, J=118).

(1) Convergence analysis: The ACO-SE has promising convergence and high convergence speed. Figure 3(a) shows, in the early stages, the ACO-SE shows a very high convergence speed, it needs a very few iterations when solution is near to the COI solution, and after 70~80 iterations, it achieves a relative stable state. In theory, when g is larger (Scenario II), there will be a more candidate SKUs set **A0** which will potentially lead to a larger solution space and a slower convergence speed. However, since the problem is small, there is no obvious difference in the convergence speed for the two scenarios.

(2) Improvement on the makespan: The ACO-SE can achieve better picking efficiency than COI. Figure 3 (a) shows that, in the Scenario-I, the improvement between ACO-SE (the makespan is 1885.2 s for 216 cartons) and COI (2307 s) is approximately 18.28%; in Scenario-II, the improvement between ACO-SE (the makespan is about 1791.2 s for 118 cartons) and COI (2205.6 s) is approximately 18.78%.

(3) Stability of solution: The ACO-SE has a stabile solution and not large fluctuations and standard deviations. We replicate the improved ACO-SE 10 times, as shown in Figure 3 (b), the average makespan is about 1887.02 s, the worst makespan (1898.1 s) is about 1.59% greater than the best (1865.3 s), the standard deviation is about 8.71 s.

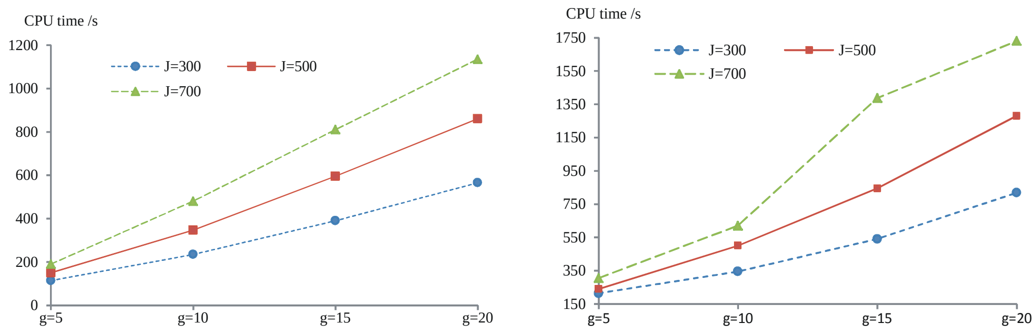
The average CPU time of 500 iterations is about 3.10 minutes, the average CPU time of finding the best solution is about 20~60 s (i.e. 50~150 iterations), it shows a preferable convergence speed.

4.3 Impact of correlation on picking efficiency for medium and large problems

4.3.1 Impact of the correlation strength

We use opposite and comparison tests to evaluate the impact of the correlation strength on the picking efficiency. In these tests, we set K=540 and F_i=0.1. 72 scenarios were constructed based on the different J, $C(i,j)$ and g values and we use the method proposed by Kim and Smith (2012) to generate 10 random problems for each scenario (i.e. 720 problems), and each problem is replicated 10 times with appropriate ACO-SE experimental parameters.

Figure 4 shows the CPU time of the ACO-SE when K=540 and K=1080 respectively. From Figure 4, it is clear that the CPU time increases as g, J and K increase; the time-difference among between scenarios with different numbers of cartons (i.e. J) will increase as the numbers of line items per carton (i.e. g) increases. Clearly, as expected, the ACO-SE will consume more time as the problem size increases. For different problem sizes, the ACO-SE only needs about 50~200 iterations to get a satisfied solution.



(a)K=540 (b) K=1080

Figure 4. CPU time when It_m=500, F_i=0.1 and $C(i,j)$=2

Tables 5 and 6 illustrate the summary of the average makespans of ACO-SE and COI.

Table 5 Solution bounds and comparison between ACO-SE and COI by different $C(i,j)$

K	J	$C(i,j)$	Idealized value range /s				ACO-SE /s				COI/s				Improvement percentage /%			
			g=5	10	15	20	5	10	15	20	5	10	15	20	5	10	15	20
		1					2658.54	5041.56	6934.22	8445.61	3700.03	6064.89	7863.95	9274.16	28.15	16.87	11.83	8.93
	300	2	886 5878	1405, 11713	1925, 12148	2443, 12583	2662.76	5035.85	6953.33	8436.22	3563.24	5890.02	7703.65	9153.23	25.27	14.50	9.74	7.83
		30					2717.67	5063.13	7002.86	8497.15	3410.65	5945.92	77085.6	9110.22	20.32	14.85	9.12	6.73
		1					4751.81	8829.12	12011.72	14477.87	5904.27	10092.71	13086.04	15499.11	19.52	12.52	8.21	6.59
540	500	2	1448, 9768	2313, 19493	3178, 20218	4043, 20943	4738.83	8832.13	12002.03	14497.58	5854.13	10029.01	12859.4	15342.12	19.05	11.93	6.67	5.51
		30					4775.85	8868.93	12109.11	14580.02	6092.23	9851.35	12788.04	15126.91	21.61	9.97	5.31	3.62
		1					6951.97	12679.58	17107.73	20526.09	8309.88	14267.41	18375.42	21215.68	16.34	11.13	6.90	3.25
	700	2	2010, 13658	3221, 27273	4432, 28288	5643, 29303	6921.18	12683.87	17027.11	20475.01	8327.46	14001.67	18267.89	21366.67	16.89	9.41	6.79	4.17
		30					6899.49	12646.44	17011.45	21025.72	8284.46	13743.76	18470.45	21073.76	16.72	7.98	7.90	0.23
		1					1433.84	2655.53	3905.14	5036.93	1890.78	3569.33	4879.56	6090.56	24.17	25.60	19.97	17.30
	300	2	464.5, 2960.5	724, 5878	983.5, 8795.5	1243, 11713	1470.92	2650.82	3913.36	5049.02	1878.46	3502.17	4977.67	6033.44	21.70	24.31	21.38	16.32
		30					1464.45	2726.01	3933.73	5058.12	1970.33	3449.89	4808.22	5990.23	25.68	20.98	18.19	15.56
		1					2394.20	4743.96	6906.18	8786.62	3215.04	5928.07	8245.04	10105.14	25.54	19.98	16.24	13.05
1080	500	2	745.5, 4905.5	1178, 9768	1610.5, 14630	2043, 19493	2404.82	4741.63	6919.81	8821.33	3003.98	5905.91	8168.56	10028.89	19.95	19.71	15.29	12.04
		30					2421.73	4730.36	6944.34	8843.36	3282.86	5897.56	8095.42	9926.71	26.23	19.79	14.22	10.91
		1					3474.64	6914.72	9955.86	12637.47	4441.37	8292.45	11382.02	13999.03	21.77	16.61	12.53	9.73
	700	2	1025.5, 6850.5	1632, 13658	2237.5, 20465.5	2843, 27273	3449.08	6936.32	9977.27	12716.42	4648.89	8170.21	11338.91	14095.02	25.81	15.11	12.01	9.78
		30					3473.48	6925.41	10056.68	12655.51	4375.67	8364.94	11291.04	13976.61	20.62	17.21	10.93	9.45

Table 6 Average improvement between ACO-SE and COI by random $C(i,j)$

K	J	ACO-SE /s				COI/s				Average improvement percentage /%			
		g=5	10	15	20	5	10	15	20	5	10	15	20
	300	2679.65	5046.84	6963.39	8459.63	3557.97	5966.94	7758.70	9179.20	24.69	15.42	10.25	7.84
540	500	4755.46	8843.36	12040.90	14518.43	5950.21	9991.02	12911.16	15322.71	20.08	11.49	6.74	5.25
	700	6924.16	12669.96	17048.61	20675.61	8307.26	14004.28	18371.25	21218.72	16.65	9.53	7.20	2.56
	300	1456.36	2677.43	3917.36	5047.96	1913.19	3507.13	4888.48	6038.07	23.88	23.66	19.87	16.40
1080	500	2406.83	4738.61	6923.41	8817.06	3167.29	5910.51	8169.67	10020.25	24.01	19.83	15.25	12.01
	700	3465.69	6925.36	9996.53	12669.77	4488.64	8275.86	11337.32	14023.55	22.79	16.32	11.83	9.65

Table 7 Solution comparison between ACO-SE and COI by different F_i

J	F_i /%	ACO-SE /s				COI /s				Improvement percentage /%			
		g=5	10	15	20	5	10	15	20	5	10	15	20
300	1	2852.31	5248.52	7190.12	8702.47	3511.26	5900.12	7705.61	9161.22	18.77	11.04	6.69	5.01
	3	2805.23	5193.41	7125.68	8623.21	3500.21	5902.46	7712.32	9156.71	19.86	12.01	7.61	5.83
	5	2711.21	5086.02	7010.02	8514.41	3498.44	5896.98	7701.33	9146.91	22.50	13.75	8.98	6.91
	10	2662.76	5035.85	6953.33	8436.22	3563.28	5890.01	7703.54	9153.32	25.27	14.50	9.74	7.83
	15	2625.93	5051.12	6940.71	8460.51	3534.57	5883.63	7707.57	9159.12	25.71	14.15	9.95	7.63
500	1	4926.77	9123.33	12291.12	14785.32	5823.34	10001.56	12825.6	15353.23	15.40	8.78	4.17	3.70
	3	4874.15	9035.23	12195.56	14694.15	5833.65	10032.45	12843.67	15332.46	16.45	9.94	5.05	4.16
	5	4799.24	8867.17	12093.44	14575.11	5835.87	10011.11	12829.18	15340.56	17.76	11.43	5.74	4.99
	10	4738.87	8832.11	12002.02	14497.51	5854.12	10029.02	12859.41	15342.22	19.05	11.93	6.67	5.51
	15	4740.22	8822.23	11971.72	14492.01	5852.73	10006.03	12841.09	15372.11	19.01	11.83	6.77	5.73
700	1	7185.67	12995.6	17392.15	20697.79	8268.17	14011.14	18115.21	21157.88	13.09	7.25	3.99	2.17
	3	7081.12	12863.4	17187.34	20500.54	8279.45	14014.32	18176.43	21204.67	14.47	8.21	5.44	3.32
	5	6955.31	12783.9	17107.03	20465.92	8258.21	14009.37	18125.45	21242.12	15.78	8.75	5.62	3.65
	10	6921.15	12683.8	17027.13	20475.13	8327.51	14001.71	18267.88	21466.67	16.89	9.41	6.79	4.62
	15	6910.79	12643.1	17068.03	20453.74	8257.78	14007.18	18311.89	21458.79	16.31	9.74	6.79	4.68

Table 8 Average improvement between ACO-SE and SA-C by random $C(i,j)$

K	J	ACO-SE /s			SA-C /s			Improvement /%		
		g=10	15	20	10	15	20	10	15	20
540	300	5046.84	6963.39	8459.63	5144.11	7282.30	8930.60	1.89	4.38	5.27
	500	8843.36	12040.90	14518.43	9847.52	13509.31	16244.36	10.20	10.87	10.62
	700	12669.96	17048.61	20675.60	14698.76	19853.22	23728.66	13.80	14.13	12.87
1080	300	2677.43	3917.36	5047.96	2961.54	4547.86	5853.09	9.59	13.86	13.76
	500	4738.61	6923.40	8817.06	5760.47	8441.87	10663.02	17.74	17.99	17.31
	700	6925.36	9996.53	12669.77	8681.68	12448.18	15719.00	20.23	19.69	19.40

In Table 5, the "Idealized value range" means the lower and upper bounds of makespan in two ideal conditions. The *Low* bound is the makespan when each carton visits only *one* zone to reduce the carton initial setup time S_t and the pickers walk the minimum distance in each zone; the *Up* bound is the make span when each carton visits as much as possible zones and pickers walk the maximum distance in each zone. The Idealized value range is independent on the $C(i,j)$.

(1) Table 5 shows, the ACO-SE yields better makespans than COI, in general. The Maximum improvement between ACO-SE and COI is about 28.15%, which means that 135.12 minutes saving can be obtained during each shift (480 minutes). In general, for the given J and $C(i,j)$, the improvement percentage decreases as g increases, because there will be more slots and zones needed to be visited when g increases gradually, which will lead to more walking distance. The further the picker must walk into the zone, the weaker the impacts of the correlation on the picking efficiency.

(2) From Table 5, for the given J and g, when J=5, there is no obvious gap between scenarios with different $C(i,j)$ values; when J is larger than 5, the improvement percentage decreases as $C(i,j)$ increases. The correlation strength $C(i,j)$ has a little impact on the picking efficiency.

Furthermore, we present the average improvement percentage of three degrees of correlation strength $C(i,j)$ in Table 6. The cell in Table 6 is indicated by a set of a level of the SKUs, a level of the numbers of cartons, and a level of the number of line-items. Form the Table 6, the ACO-SE always provides better makespan than COI, the average improvement percentage between ACO-SE and COI is shown from 2.56%~24.69%. In general, the average improvement percentage decreases as the numbers of line-item (g) is large, or as the numbers of cartons (J) is large, or as the numbers of SKUs (K) is small.

(3) From Table 5, in general, there is no obvious gap among different degrees of the $C(i,j)$, the correlation strength has no obvious impact on the picking efficiency, this is because the correlated SKUs just needed to assigned to the same zone, no need to be assigned to the adjacent locations, this rule is different from Kim and Smith(2012) who concluded that the more correlation strength, the more picking efficiency. With the return travel policy, one of correlated SKUs can be allocated to any locations along the path from the initial point to the other SKU's location.

The makespan increases as the numbers of carton (J) and line-items (g) are large. The makespan decreases as the numbers of zone (M) and the numbers of SKUs (K). When K increases, the space of the fast pick area will be enlarged by the following ways: to increase the number of zones and pickers, which will increase the space and labor cost; or to increase the slots in zones by increasing the rack length or/and the rack levels, which will increase the makespan and facilities cost accordingly. Thus, there is a trade-off between time and costs to decision how to adjust the fast pick area.

4.3.2 Impacts of the correlation probability

In this part, we will use the opposite and comparison test to evaluate the impacts of the correlation probability (F_i) on the picking efficiency. The correlation probability F_i is defined as the proportion of correlated SKUs with SKU *i* to the total SKUs, it shows that how many SKUs are correlated with SKU *i*.

We set $C(i,j)$=2, K=540, and do some tests on different J, F_i and *g*. Totally, 60 scenarios are constructed randomly based on the different J, F_i and *g*, we use the same method (Kim and Smith, 2012) to generate 10 problems for each scenario randomly (i.e.600 problems totally), each problem is replicated 10 times randomly. Table 7 illustrates the test results.

From the Table 7, the ACO-SE always provides better makespan than COI, the improvement percentage between ACO-SE and COI is shown from 2.17%~25.71%.In general, for the given J and F_i, the improvement percentage decreases as *g* becomes greater, this trend is as same as the previous test results.

When F_i is less than 10%, the improvement increases as F_i becomes greater; when F_i is greater than 10%, the improvement has no obvious difference for the different F_i. This is because, in zone-based picking system with return travel policy, there are totally 54 SKUs in each zone, so when F_i is greater than 10%, the more correlated SKUs have no chance to be assigned to the same zone. We can conclude that the correlation probability has significant impacts on the picking efficiency, which is different from the correlation strength $C(i,j)$ which has no obvious impacts on the picking efficiency, when there are more correlated SKUs, there will be more improvement potentially.

4.3.3 Comparison with SA-C heuristic

Kim and Smith (2012) proposed a Simulated Annealing algorithm using correlation interchange for dynamic which was called SA-C heuristic. Based on the idea that SKUs that appear together in the same carton should be located near each other in the picking area, SA-C uses the information from the correlated list and performs correlated interchange randomly. When there is some improvement, SA-C will keep the interchange, otherwise, in order to escape the local optima, SA-C will accept the non-improving interchange at some probability hoping to expand the search space and ultimately reach a better overall solution. By the comparison between the ACO-SE and SA-C, there are some improvement on the CPU time and makespan:

(1) For the CPU time, when K=540, J=300 and *g*=10~20, the ACO-SE only needs about 20~90 s (which varies based on the size of the problems) to get a satisfied solution, but the SA-C needs about 600~2000 s to get a satisfied solution, it means that the ACO-SE has better convergence speed than SA-C heuristic for the same size problem and needs fewer CPU time. The reason is, in the each of the iteration, the ACO-SE can use lots of correlation information to do some arrangements for all SKUs, but the SA-C can only use one piece of correlation information and do at most one slot exchange.

(2) Table 8 illustrates the comparison result between ACO-SE and SA-C heuristic when F_i=10%, $C(i,j)$=1,2,30 at the same probability. It shows that on all scenarios, the ACO-SE can reach a better solution than SA-C heuristic for the same size problem and the average improvement varies from 1.89%~20.23%. This is because, the SA-C just put the correlated SKUs to the adjacent slots, but in our picking system with Return travel policy, the “proximity” of the SKUs in the same zone is not important: they just need to be assigned to the same zone. One of correlated SKUs

can be allocated to any locations along the path from the initial point to the other SKU's location. So we use slots-exchange policy to ignore the proximity and achieve a better picking efficiency. The comparison results prove that the "proximity" of the SKUs is not important.

It is clear that the average improvement becomes large, as the number of cartons (J) is large, number of SKUs (K) is large; but it is difficult to find a consistent trend by changing the number of line-items (g).

5 Conclusions

We believe that the slotting problem based on correlations is a fundamental, but curiously overlooked problem in warehouse operation optimization. The problem in this study is the dynamic slotting problem for a pick-wave zone-based picking order system given various scenarios of Cartonization with SKUs correlation. Through our research we have focused on the correlation strength and probability among SKUs in a pick wave. The problem is NP-hard and the size of a real problem is very large, we proposed an ACO with slot-exchange policy to solve the MIP model from Kim and Smith [16], and used three methods to test the performance and evaluate the impacts of the SKUs correlation on the picking efficiency. Some promising results are given as follows:

(1) The ACO-SE shows promising convergence, makespan improvement, stability solution and preferable computing speed (CPU time). For two small problems, the improvement on makespan between ACO-SE and COI is about 18.28% and 18.78% respectively. The solution of ACO-SE has no large fluctuations and standard deviations; the average CPU time of finding a stability best solution is about 20~60 s.

(2) For the medium and large problems, the ACO-SE provides better makespan than COI. The correlation strength has no obvious impacts on the picking efficiency, but the correlation probability has some significant impact one the picking efficiency, when there are more correlated SKUs, there will be more improvement potentially. For the average improvement, the ACO-SE always provides better makespan than COI too; the average improvement percentage varies from 2.17%~25.71%.

(3) By the comparison with the SA-C heuristic, the ACO-SE has better CPU time and convergence speed and can achieve a better makespan, the average improvement varies from1.89%~20.23%, which proves that the "proximity" of the SKUs is not important, they just need to be assigned to the same zone. With the return travel policy, one of correlated SKUs can be allocated to any locations along the path from the initial point to the other SKU's location.

The best slotting depends on how to assign orders to cartons given the numbers of orders in a pick wave (i.e. Cartonization) and the best Cartonization depends on how to assign SKUs to slots (i.e. slotting). The limitation in this paper determines the slotting problem given Cartonization information. It is clear that slotting and Cartonization problems affect each other. In the further study, we expect that a potential improvement can be obtained by considering the two correlated problems systematically.

References

[1] Bartholdi, J.J., Hackman, S.T., 2008.Allocating space in a forward picking area of a distribution center for small parts. *IIE Transaction, 40,* 1046-1053

[2] Bie W Q, Li Y J, 2004.Dynamic location assignment of AS/RS based on genetic algorithm.*Computer Engineering and Applications*, *45*(29),211-219

[3] De Koster, R.,Le-Duc,T., Roodbergen,K.J.,2007.Design and control of warehouse order picking: A literature review. *European Journal of Operation Research,* 182(2), 481-501

[4] Frazelle, E.H., Sharp, G.P., 1989.Correlated assignment strategy improve order-picking operation. *Industrial Engineering*, 4, 33-37

[5] Frazelle, E.H., 1990. Stock location assignment and order picking productivity, MHRC-TD-89-11, *Material Handling Center*, Atlanta, Georgia

[6] Frazelle, E.H., Hackman, S.T., Passy, U., 1994. The forward-reserve problem.In: Ciriani, T.C., Leachman, R.C (Eds), *Optimization in Industry2*, Weiley, New York, 43-61

[7] Frazelle, E.H., 2002. World-class warehouse and material handling. NY: McGraw Hill

[8] Goetchslckx M., Askhyeri,J., 1989. Classification and design of order picking systems. *Logistical World*, (6),99-106

[9] Goetschalckx, M. and Ratliff, H.D., 1990. Shared storage policies based on the duration of stay. *Management Science,* 36(9), 1120-1132

[10] Harmatuck, D.J., 1976. A comparison of two approaches to stock location. *Logistics and Transportation reviews*, 12(4), 282-284

[11] Hackman, S.T. and Platzman, L.K., 1990.Near optimal solution of generalized resource allocation problems with large capacities. *Operations Research*, 38(5), 902–910

[12] Heskett, J.L., 1963. Cube-per-order index- A key to warehouse stock location. *Transport and Distribution Management,* 3,27-31

[13] Heskett, J.L., 1964. Putting the Cube-per-order index to work in warehouse layout. *Transport and Distribution Management*, 4,23-30

[14] Hwang, H., Oh, Y.H., and Cha, C.N., 2003. A performance evaluation for a low level picker-to-part system. *Engineering optimization*, 35,285-295

[15] Kim, B.S., Smith, J.S., 2008(a). Dynamic slotting for zone-based distribution center picking operation[A].*10th International Material Handing Research Colloquium*, Dortmund,Germany,559-599

[16] Kim, B.S., Smith, J.S., 2012. Slotting methodology using correlated improvement for a zone-based carton picking distribution system. *Computers &*

Industrial Engineering, Vol. 62, Issue 1, pp. 286-295

[17] Landers, T.L. and Beaver, M.K., Sadiq, M., and Stuart, D.E.,1994. Software for dynamic re-configurable order picking systems. *Computers & Industrial Engineering*, 1-4, 245-248

[18] Liu S.N., Ke Y.L., Li J.X., 2006. Optimization for automated warehouse based on scheduling policy [J]. *Computer Integrated Manufacturing Systems*, 12 (9), 1438 -1443

[19] Malmborg, C.J. and Bhaskaran, K., 1987. On the optimality of the cube per order index for conventional warehouses with dual command cycles. *Material Flow*, 4, 169–175

[20] Malmborg, C.J., 1995. Optimization of Cubic-per-Order Index layouts with zoning constraints. *International Journal of Production Research*, 33(2), 465–482

[21] Manzini, R., 2006. Correlated storage assignment in an order picking system. *International Journal of Industrial Engineering*, 13(4), 384-394

[22] Petersen, C.G., 1997. An evaluation of order picking routing policies. *International Journal of Operations & Production Management*, 17(11), 1098–1111.

[23] Petersen, C.G., and Aase, G., 2004. A comparison of picking, storage, and routing policies in manual order picking. *International Journal of Production Economics*, 92, 11 –19.

[24] Sadiq, M., Landers, T.R., and Taylor, G.D., 1996. An assignment algorithm for dynamic picking systems. *IIE Transactions*, 28(8), 607-616

[25] Sahni, S. and Gonzalez, T.L., 1976. P-complete approximation problems. *Journal of the association of computing machinery*, 23,555-565

[26] Van den Berg, J.P., Sharp, G.P. Forward-reserve allocation in a warehouse with unit-load replenishment. *European Journal of Operation Research*, 1998, 111:98-113

[27] Xiao J, Zheng L., 2008. Storage location assignment in a multi aisle warehouse considering demand correlations. *Computer Integrated Manufacturing Systems*, 14(12), 2447-2451

[28] Zhang Y F, Bo L., 2004.Application of Genetic Algorithm in Selecting Accurate Freight Site. *Journal of System Simulation*, 16(1), 168-171.