

Cartonization and Slotting Optimization Model and Algorithm for Considering Correlation in Hybrid Travel Policy

Yao Yuepeng

IT Department

Zhejiang Red Dragonfly shoes Corporation Limited
Wenzhou, P.R. China

Li Yingde

Institute of industrial engineering
Zhejiang University of technology
Hangzhou, P.R. China
liyingde2008@sina.com

Zhou Xiaomin

Faculty of Mechanical Engineering & Automation
Zhejiang Sci-Tech University
Hangzhou, P.R. China

Ning Fanghua

Faculty of Mechanical Engineering & Automation
Zhejiang Sci-Tech University
Hangzhou, P.R. China

Abstract—In wave-picking and entire-replenishment zone-based warehouse, the slotting and cartonization problems occur with equal frequency and can be the optimization conditions for each other. A dynamic location assignment model is proposed to minimize the wave-picking time among all of zones, and a particle swarm optimization (PSO) for slotting is designed according to SKUs correlation and simulate cartonization optimization (SCO) based on SKUs' location. Therefore, the new PSO-SCO heuristic algorithm is given to solve the slotting and cartonization problem simultaneously. With the improvement of picking time is from 2.6% to 8.8%, results show that the solution of PSO-SCO algorithm is better than COI and PSO-SCO algorithm. It has significant impact on improving the picking efficiency.

Keywords—Slotting optimization; Cartonization optimization; PSO; SKUs correlation; Collaborative optimization algorithm

I. INTRODUCTION

On one hand, SKU correlation is determined by the result of cartonization optimization, and location assignment based on SKU correlation can reduce picking walking time; on the other hand, location assignment determines the distance between SKUs, and cartonization optimization based on the SKUs position can improve picking efficiency. In view of this, this paper puts forward collaborative optimization program to solve cartonization and location assignment problem simultaneously by making them to be the optimization conditions each other and then optimizing it for the shortest picking time.

II. BACKGROUND

De Koster et al[1] think that order picking accounts for over 40% of the total picking time and over 65% of the

total operating cost. Therefore, reasonable picking optimization plays an important role in improving the efficiency of order picking, and the commonly method of it is slotting and cartonization optimization.

Slotting optimization is to determine SKUs storage location for the highest picking efficiency. Traditional studies[2-3] of slotting optimization focus on the SKUs shelf or their characteristic, without considering about SKUs correlations. Yet, COI advanced by Heskett[4-5] cannot get better optimization scheme in the case of SKU correlation.

The gist of exploiting the SKU correlations during location assignment is to assign them to the adjacent position. Slotting optimization problem was solved by use of historical statistical correlation in much previous researches. The patterns of SKUs flow always changes dynamically or periodically due to various factors. For example, the slotting would change with the highly dynamic demands of customers in B2C, so that historical statistical correlation cannot accurately reflect the SKU correlations. Therefore, Kim and Smith[6] found a way to generate SKUs correlation data, and proposes a slotting optimization algorithm which achieve well results.

Cartonization optimization is to pack SKUs in order into carton. It aims to minimize the cartons and picking walking time[1,6]. Orders are boxed in batches when requiring multiple cartons[7,8]. After that, assigning SKUs which in the same carton to the adjacent location would reduce picking walking time in possible. Bozer[9] proposed the more SKUs each box can be loaded, the shorter walking time is:

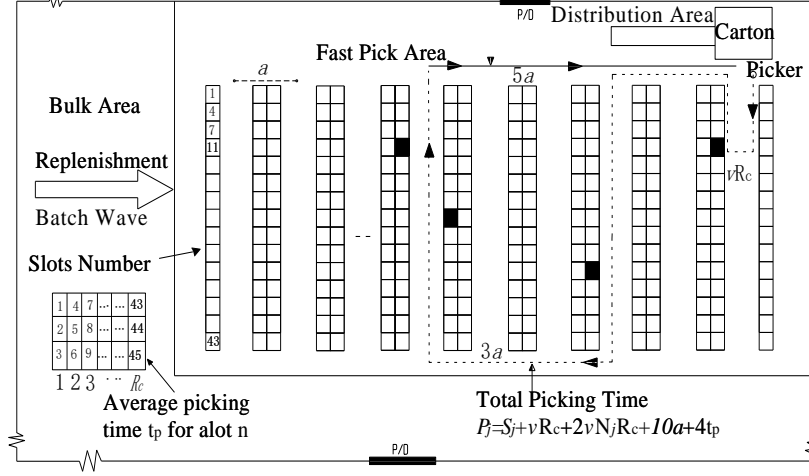


Figure 1. Configuration of dynamic pick-wave zone-based warehouse

III. SLOTTING OPTIMIZATION MODEL

To attain the object of minimum picking time t_{pm} , we set up the slotting optimization model as (1):

$$t_{pm} = \sum_{j=1}^J (E_{zqj} E_{zlj} H_j R_c v + 2(Z_j - 1) t_a + 2E_{zqj} E_{zlj} N_{zjR_c} v) + \sum_{j=1}^J \sum_{k=1}^K \sum_{z=1}^Z \sum_{x=1}^X n_{kzx} C_{ijk} t_p \quad (1)$$

$$\text{Then the objective function as (2): } \text{Min Max}(t_{pm}) \quad (2)$$

$$\text{Constraint conditions: } \sum_{z=1}^Z \sum_{x=1}^X n_{kzx} = 1, \forall k. \quad (3)$$

$$\sum_{k=1}^K n_{kzx} \leq 1, \forall z, x. \quad (4)$$

$$\sum_{z=1}^Z G_{jz} \geq 1, \forall j. \quad (5)$$

$$\sum_{i=1}^I U_{ij} \geq 1, \forall i. \quad (6)$$

$$\sum_{i=1}^I U_{ij} \leq 1, \forall j. \quad (7)$$

$$\sum_{j=1}^J C_{ijk} = Q_{ik}, \forall i, k \quad (8)$$

$$\sum_{i=1}^I \sum_{k=1}^K V_k C_{ijk} \leq V, \forall j \quad (9)$$

Where, J represents carton number, Z represents tunnel number; X represents slotting number, K represents the number of SKUs, M represents zoon number, V represents the volume of carton, Q_{ik} represents the total number of SKUs in order i , I represents order number, v is the picker walking speed, H_j represents total tunnel number which carton j need to access, Z_j represents tunnel number which the last carton need to access, N_{zjR_c} represents column number of the last SKU which in carton j , R_c represents column number of shelf, C_{ijk} represents the number of SKU k in order i which is assigned to carton j , Z_j represents tunnel number which the last carton need to access, t_p is the picking time of singal SKU, t_a is the walking time from a tunnel exit to the adjacent tunnel entrance.

State variables: if carton j need to access tunnel z , $G_{jz}=1$, else $G_{jz}=0$; if order i is assigned to carton j , $U_{ij}=1$, $U_{ij}=0$; if carton j cross the middle of tunnel z , $E_{zqj}=1$, else $E_{zqj}=0$; if carton j donnot need to cross the middle of tunnel z , $E_{zlj}=1$, else $E_{zlj}=0$.

Decision variables: $n_{kzx}=1$, if SKU k is assigned to position x in tunnel z , else $n_{kzx}=0$.

IV. COLLABORATIVE OPTIMIZATION ALGORITHM FOR SLOTTING AND CARTONIZATION

If the information of cartonization and location is uncertain, we can assume the information of cartonization and location is known. We design slotting optimization algorithm based on SKUs correlation and cartonization optimization algorithm based on location, and then propose collaborative optimization algorithm in the use of nested iterative method to solve slotting and cartonization problem simultaneously.

A. Slotting optimization algorithm

The slotting optimization algorithm is designed according to simulating the flock foraging behavior, each SKU can be viewed as a particle with no quality and volume in the optimization space, and the position of each particle represents the location assignment. The particle updates it's location with a corresponding speed which is based on the SKUs correlation strength. SKUs correlation strength can be represented by correlation weight $C(i,j)$, which is the frequency of two SKU in the same carton in a picking wave. The greater the $C(i,j)$ is, the stronger the correlation between SKU i and j .

For a given SKUs location set, we generate a set of SKUs location encoding; initialize the current zone index $m=1$; let slot 1 as the current slot, set the slot index $n=1$; and mark the SKU in the current slot as SKU i , other SKUs as SKU j . the steps of updating each SKU location are as following:

1) According to correlation information list L , generate a set $P0 = \{j | C(i,j) > 0, i, j \in P\}$, in which SKUs are correlated with current SKU i , the remaining SKUs are in set $P1 = \{j | C(i,j) = 0, i, j \in P\}$;

2) Initialize each particle's velocity vector, if $j \in P0$, $V_j = -1$ which means to move forward a location actively; else $j \in P1$, $V_j = 0$ which means the location doesn't change;

3) Initialize the particle swarm, using COI method, to generate initial particle swarm location, and then to set $k=0$, remember current location encoding as S^0 , the size of location is N ;

4) Calculate the fitness of every particle for the particle swarm in generation k . the fitness function is as (10):

$$\text{fit}_j = C(i,j) / |X_i - X_j| \quad (10)$$

Then Calculate the local optimal position of every particle (i.e. p_j) and the overall optimal position of particle swarm (i.e. g_b);

5) Update particles' velocity according to (11)

$$V_j^{k+1} = V_j^k + C_1 R_1(p_j - X_j^k) + C_2 R_2(g_b - X_j^k) \quad (11)$$

Where $C_j = C(i,j)$ means the particle's cognitive ability is the correlation among the other SKU, which can avoid algorithm to fall into local optimum; $C_2=1$ is to ensure certain social information sharing between particles, which can speed up the algorithm searching; $R_1, R_2 \sim U(0,1)$ is a random variable;

6) Update each particle's position, if $P0 = \phi, P1 \neq \phi$, randomly transforms location encoding of the SKUs; if $P0 \neq \phi, P1 \neq \phi$, to transform SKUs' location encoding according to (12) in set $P0$, and SKUs in set $P1$ are assigned to the spare location in turn and get new location encodings;

$$X_j^{k+1} = X_j^k + V_j^{k+1} \quad (12)$$

Where, X_j^{k+1} and X_j^k are the location encodings of SKU j , V_j^{k+1} is the velocity of SKU j .

7) Sort the updated SKUs location encoding in ascending; if the SKUs' location encoding are identical, they are sorted by $C(i,j)$, the larger one in the front; therefrom we generate a set of slot sequence, and get a set of feasible solution S^k ;

8) Turn to step h if the iterative condition is met; else $k=k+1$, turn back to step d ;

9) Output current location of SKUs, which will be the optimal location allocation.

they are sorted by $C(i,j)$, the larger one in the front; therefrom we generate a set of slot sequence, and get a set of feasible solution S^k ;

10) Turn to step h if the iterative condition is met; else $k=k+1$, turn back to step d ;

11) Output current location of SKUs, which will be the optimal location allocation.

B. Simulate containerization optimization

he number of carton is related with SKU's volume in cartonization problem, therefore the main problem is to

optimize the picking walking distance. We assume that location information is known, design SCO according to the location position encoding:

2) To sort the order according to time sequence, the order set is $I = \{i | i=1, 2, \dots, I\}$, set $i=1$;

3) Let order i as the current order, to arrange SKUs of order i in ascending order according to their position encoding;

4) Packing SKUs into carton j according to its sequence;

5) Each time packing in a SKU, to judge the remaining space of the carton, if enough, to pack in another SKU; else to close current carton and open a new carton $j+1$ as the current carton. Repeat this step until all of SKUs in order i are packed into the carton, then set $i=i+1$, turn to step 1 .

C. Collaborative optimization algorithm

We design collaborative optimization by using the nested iterative method. The steps are as follows:

1) Using COI to generate initial location encoding, remember it as S^0 , set $S=S^0$;

2) According to location information provided by S , getting a solution of containerization problem by running SCO, to remember it as B ;

3) According to containerization information provided by B , to generate correlation information list L , getting the optimal solution S^k by running PSO;

4) If it meets the iteration termination condition, the optimal solution of PSO-SCO will be S^k and B ; else to set $S=S^k$ and turn to step b .

V. ALGORITHM TEST AND ANALYSIS

We develop PSO-SCO in VB.NET2008 and test it in Windows 7 64-bit, 8GB of RAM and Intel Core TM i5-520 (2.4 Ghz) CPU.

Situation 1: Randomly generating 300 orders, each order randomly contains 10 kinds of SKUs, carton's capacity is 1, 2, 3, 4, 5, 10, 15, 20 SKUs respectively, and to compare the result of COI, SAC-NFDP [10] and PSO-SCO. Each Algorithm runs 10 times, the result is as Table I:

Situation 2: When size of order changes, to compare different performance by different order size G and different carton's capacity g . The result is as Fig .2:

TABLE I SOLUTION BOUNDS AND COMPARISON BETWEEN ALGORITHMS BY DIFFERENT CARTON CAPACITY

	g=1	g=2	g=3	g=4	g=5	g=10	g=15	g=20
picking time of COI/s	4832.5	3329.7	2789.1	2274.5	1675.3	1369.6	1369.6	1369.6
picking time of SAC-NFDP /s	4832.5	3159.7	2689.1	2174.5	1615.3	1329.6	1329.6	1329.6
picking time of PSO-SCO/s	4832.5	3036.6	2591.4	2091.4	1561.4	1295.5	1295.5	1295.5
Improvement percentage compare to COI	0%	8.8%	7.1%	8.1%	6.8%	5.4%	5.4%	5.4%
Improvement percentage compare to SAC-NFDP	0%	3.9%	3.6%	3.8%	3.3%	2.6%	2.6%	2.6%

1) As Table 1 shows, when carton's capacity $g>1$, the result of PSO-SCO is about 5.4%~8.8% better than COI, and about 2.6%~3.9% better than SAC-NFDP; the picking time and improvement percentage decrease as g increases.

2) Fig .2 shows that the bigger G is, the larger target value is. It indicates that with the increasing of G , more SKUs need to be picked, and the more picking time is required. In the same time, with the increasing of g , the

target value becomes small and the improvement decreases.

3) Fig .2 shows, when $G > 1$, $g=1$, the target value is larger, because there is no correlation between SKUs when $g=1$.

4) Fig .2 indicates, when $G < g$, a single carton can meet the picking demand, target value remains unchanged; when $G = g$, target value starts to increase; when $G > g$, the picker needs to get a new carton, then picking walk time increases, target value increases gradually.

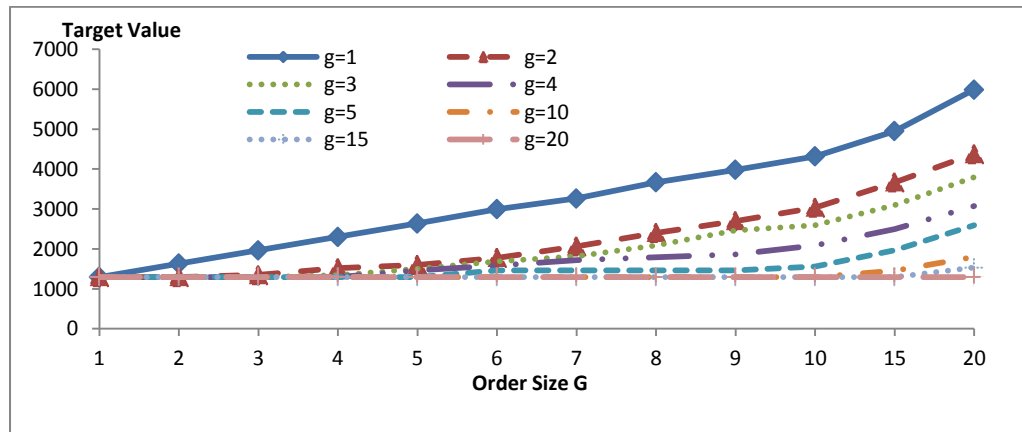


Figure 2. Performance of PSO-SCO by different G and g

VI. CONCLUSIONS

Using hybrid picking strategy, this paper built a dynamic location assignment model to minimize the pick wave makespan, designed collaborative optimization algorithm to solve slotting and cartonization problem simultaneously. Some promising results are given as follows: *a.* PSO-SCO provides better solution quality. The walking picking time has improved 5.4%~8.8% compared to COI, improved 2.6%~3.9% compared to SAC-NFDP, and improvement decreases with the increasing of carton's capacity; *b.* In wave-picking and entire-replenishment zone-based warehouse, it can improve picking efficiency that make cartonization and location assignment problem to be the optimization conditions each other.

VII. ACKNOWLEDGMENT

This research was supported by Zhejiang provincial National Science Foundation of China under Grant No. LY13G020019 and by Zhejiang University National Science Foundation under No. 2012XZ001

REFERENCES

[1] De Koster, René Tho Le-Duc, and Kees Jan Roodbergen. "Design and control of warehouse order picking: A literature review." *European Journal of Operational Research*, vol. 182, no. 2, pp. 481–501, 2007.

[2] Brynzák, Henrik, and Mats I. Johansson. "Storage location assignment: Using the product structure to reduce order

picking times." *International Journal of Production Economics*, vol. 46, pp. 595-603, 1996.

[3] Muppani, Venkata Reddy, and Gajendra Kumar Adil. "A branch and bound algorithm for class based storage location assignment." *European Journal of Operational Research*, vol. 189, no. 2, pp. 492-507, 2008.

[4] Heskett, James L. "Cube-per-order index—a key to warehouse stock location." *Transportation and distribution Management*, vol. 3, no. 1, pp. 27-31, 1963.

[5] Heskett, James L. *Business logistics: management of physical supply and distribution*. Ronald Press Co., 1964.

[6] Kim, B. S., and J. S. Smith. "Dynamic slotting for zone-based distribution center picking operation." *10th International material handling research colloquium*, Dortmund, Germany. 2008.

[7] Henn, Sebastian, and Verena Schmid. "Metaheuristics for order batching and sequencing in manual order picking systems." *Computers & Industrial Engineering*, vol. 66, no. 2, pp. 338-351, 2013.

[8] Bonassa, Antonio Carlos, and Claudio Barbieri da Cunha. "The order-picking routing problem for low-level order picker in a warehouse." *Gestão & Produção*, vol. 18, no. 1, pp. 105-118, 2011.

[9] Bozer, Yavuz A., and Justin W. Kile. "Order batching in walk-and-pick order picking systems." *International Journal of Production Research*, vol. 48, no. 7, pp. 1887-1909, 2008.

[10] Li Yingde. "Model and algorithm for cartonization and slotting optimization simultaneously in wave-picking zone-based system". *Systems Engineering — Theory & Practice*. vol. 33, no. 5, pp. 1269, 2013.