Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 83 (2019) 513–516

www.elsevier.com/locate/procedia

11th CIRP Conference on Industrial Product-Service Systems

A dynamic programming approach for storage location assignment planning problem

Meng Wang[a], Ren-Qian Zhang[a,*]

[a]*School of Economics and Management, Beihang University, Beijing 100191, China*

Abstract

Online retailers usually provide fast product delivery service to attract consumers and in consequence boost their sales, which enforces warehouses or distribution centers to fulfill customer orders within tight time windows. This paper presents a novel approach to improve the order-picking operation which is the most time- and labor-intensive activity in the process of order fulfillment. We are motivated by the fact that many product demands fluctuate over time and it would be benefic to update storage location assignment in time to reassign the most popular products in each period to storage locations nearest to the depot. We formulate the problem as an integer programming and develop a dynamic programming approach to solve it. To evaluate the proposed method, an computational experiment is conducted and the results are reported.

Peer-review under responsibility of the scientific committee of the 11th CIRP Conference on Industrial Product-Service Systems

Keywords: Order-picking; Storage location reassignment; Dynamic programming; Logistics

1. Introduction

According to the National Bureau of Statistics of China, in 2017 the Chinese e-commerce reached to 29.16 trillion CNY with a rate of 11.7%. Meanwhile, the online retail sales increased 32.2% and reached 7.18 trillion CNY. The rapid development makes online retail worthwhile to study. One important service provided by online retailers is the product delivery service (PDS) which delivers the online ordered products to customers. In recent years, people are becoming more and more accustomed to shopping on the Internet and expect their goods can be delivered as soon as possible. A higher PDS efficiency usually attracts more consumers and therefore results in higher sale volume. For example, the fast PDS that ensures customers can receive their goods before the next 11 o'click (a.m. or p.m.), has boosted the sales of JD.com, one of the largest e-commerce companies in China. To achieve high PDS efficiency, warehouses or distribution centers must be efficiently operated so that customer orders can be fulfilled in tighter time windows, among which order-picking is the most important activity.

Order-picking is the process of retrieving requested items from their storage locations to fulfill specific customer orders, which is the most time-consuming and labor-intensive operation in warehouse. According to [5], order-picking accounts for about 35% of warehouse operation expense. Common ways to improve order-picking efficiency include: (1) assigning storage locations to appropriate items (*storage policy*), (2) determining appropriate route of picking tour (*routing*), and (3) picking several orders in a batch (*batching*) [1, 3]. Instead of the aforementioned approaches, this study improves the order-picking efficiency in a novel way.

Storage location assignment (SLA) is a basic configuration for order-picking improvement, since many other approaches, such as: routing and batching, have to be done with SLA information. In the past decades, many storage policies are proposed, including: *random*, *class-based* [2, 7], *dedicated* [4] and *family-based* [6, 8] storage policies. Most storage policies or methods focus on producing an efficient SLA, minimizing the total travel distance or picking time in certain period. When planning horizon is long, however, many item demands fluctuate over time and it is necessary to update the SLA in time to reassign the most popular items in each period to the most valuable storage locations (near to the depot), so that the order-picking efficiency is improved. For example in clothes warehouses, the padded jacket products are lowly (highly) demanded in summer (winter) days and should be placed far from (near to) the depot for easier access. In reality, the reassignment decisions are made by intuitions or experiences, such as: reassigning every week, month, quarter or half-year. To the best of our knowledge, most

* Corresponding author.
E-mail address: zhangrenqian@buaa.edu.cn (Ren-Qian Zhang).

2212-8271 © 2019 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 11th CIRP Conference on Industrial Product-Service Systems.
10.1016/j.procir.2019.04.113

SLA studies pay little attention to the demand fluctuation in a long planning horizon and our motivation is to solve this research gap.

Reminder of the paper is organised as follows. We formulate SLAPP in section 2 and develop a dynamic programming approach in section 3. Section 4 presents a computational experiment and section 5 concludes the paper.

2. Problem formulation

Storage location reassignment is the process of updating SLA for specific demands. Due to the demand fluctuation over time, the reassignment could allocate popular items in each period to valuable storage locations, resulting in higher order-picking efficiency. Since storage policy is intensively studied, this paper focuses on when to reassign the storage locations for cost reduction. We refer to the problem as the storage location assignment planning problem (SLAPP).

Consider a long planning horizon in a warehouse. The planning horizon is split into T periods, at the beginning of which the storage location reassignment is possible. Given reassignment cost C, the objective is to find a solution (reassignment plan) that determines when to reassign, to minimize costs of order-picking and reassignments..

We assume that the reassignment can only be possible at the beginning of each period. This is without loss of generality, since one period can be split into two new periods if a reassignment can be conducted within it. We also assume that the items' demand information (customer orders in this paper) within each period is known and can be utilized to produce SLA by certain SLA method, such as: dedicated, class-based, or family-based storage policy. It should be noted that SLAPP is independent of order-picking systems.

The follows notations are used through out the paper:

Nomenclature

- T Number of periods in the planning horizon, indexed by $t = 1, ..., T$;
- A_t SLA that is adopted in period t;
- A_0 Initial SLA;
- O_t Customer order set in period t;
- C Travel distance for one reassignment;
- $M(O)$ Employed storage policy that output a SLA based on order set O;
- $L(A, O)$ Travel distance for picking all orders in O under SLA A;
- x_t Decision variable. $x_t = 1$, if a storage location reassignment is conducted at the beginning of period t; 0, otherwise. Let $\mathbf{x} = (x_1, ..., x_T)$.

For SLAPP formulation, we define the concept of *block*, which refers to a period sequence that starts from a reassignment period (inclusive) to the next reassignment period (exclusive). Let B be the number of blocks for solution $\mathbf{x}$, therefore

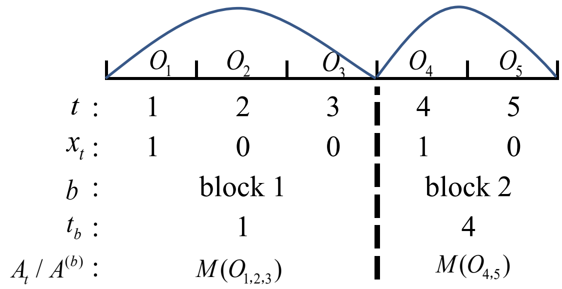

Fig. 1. Model illustration.

$B = \sum_{t=1}^{T} x_t$. Let $b = 1, ..., B$ be the block index and t_b the first period index in block b. SLAPP is modelled as the integer programming (1), which minimizes the total costs (TC) of order-picking and reassignments.

$$\min_{x_t \in \{0,1\}} TC(\mathbf{x}) = \sum_{b=1}^{B} [L(A^{(b)}, O^{(b)}) + C], \tag{1}$$

where $A^{(b)} = M(O^{(b)})$ is the SLA that is adopted in block b and $O^{(b)} = \cup_{i \in \{t_b, ..., t_{b+1}-1\}}$ the set of orders in block b. For ease of notation, let $O_{t,...,t'} = \cup_{i \in \{t,...,t'\}} O_i$ in the following sections. According to definition of block, periods from one block share the same SLA, i.e. $A_{t_b} = ... = A_{t_{b+1}-1} = A^{(b)}, \forall b = 1, .., B$.

Fig. 1 gives an example for illustration, where $T = 5$ and $\mathbf{x} = (1, 0, 0, 1, 0)$. According to the solution, we should apply reassignments in the beginning of period 1 and 4. There are $\sum_{t=1}^{T} x_i = 2$ blocks, i.e., periods 1-3 and 4-5. According to the definition of block, we have $A_1 = A_2 = A_3 = A^{(1)} = M(O_{1,2,3})$ and $A_4 = A_5 = A^{(2)} = M(O_{4,5})$. We can calculate TC as: $TC(\mathbf{x}) = L(A^{(1)}, O_{1,2,3}) + L(A^{(2)}, O_{4,5}) + 2C$.

3. Solution

The SLAPP model (1) is a binary integer programming, where $A^{(b)}$, $L(\cdot, \cdot)$ are expressed as implicit functions and make the model difficult to solve directly. Given T, there are a total of 2^T possible values for solution $\mathbf{x}$, which makes the exhaustive search method invalid. Exploiting the model characteristics, we found that the optimal solution for the first t periods can be easily determined with optimal solutions for periods before t. Therefore, an efficient dynamic programming approach (DPA) is presented to solve the SLAPP in polynomial computational efforts.

Let $F(t)$ be the optimal objective value of the first t periods. For $t = 1$, only the first period needs to be considered and there exist two strategies: reassign or not. Therefore,

$$F(1) = \min \left\{ \begin{array}{ll} L(A_1, O_1), & A_1 = A_0 \\ L(A_1, O_1) + C, & A_1 = M(O_1) \end{array} \right\}. \tag{2}$$

Since $F(1)$ has been determined, only four cases need to be examined for $F(2)$:

$$F(2) = \min \left\{ \begin{array}{ll} L(A_2, O_{1,2}), & A_2 = A_1 = A_0 \\ L(A_2, O_{1,2}) + C, & A_2 = A_1 = M(O_{1,2}) \\ F(1) + L(A_2, O_2), & A_2 = A_1^* \\ F(1) + L(A_2, O_2) + C, & A_2 = M(O_2) \end{array} \right\}, \tag{3}$$

where $A_t^* = \arg\min_{A_t}\{F(t)\}$ is the best SLA that is adopted in period t, minimizing the costs for the first t periods.

When $t = n$, the optimal strategy can be determined with the value of $F(1), ..., F(n-1)$.

$$F(n) = \min \left\{ \begin{array}{ll} L(A_n, O_{1,...,n}), & A_n = ... = A_0 \\ L(A_n, O_{1,...n}) + C, & A_n = ... = A_1 = M(O_{1,...n}) \\ F(1) + L(A_n, O_{2,...,n}), & A_n = ... = A_1^* \\ F(1) + L(A_n, O_{2,...,n}) + C, & A_n = ... = A_2 = M(O_{2,...,n}) \\ ...... & \\ F(n-1) + L(A_n, O_n), & A_n = A_{n-1}^* \\ F(n-1) + L(A_n, O_n) + C, & A_n = M(O_n) \end{array} \right\}. \quad (4)$$

When $t = T$, we get the optimal solution $F(T)$. Optimal SLA for each period can be obtained by backtracking,

$$A_t^* = \arg\min_{A_t}\{F(t)\},\ t = T, ..., 1, 0. \quad (5)$$

The computational efforts of DPA is evaluated as follows. For period t, there are $2t$ cases to be considered and therefore a total of $\sum_{t=1}^{T} 2t = T^2 + T$ cases need to be examined, which results in polynomial computational efforts. It should be noted that the computational efforts of $M(\cdot)$ and $L(\cdot,\cdot)$ that are dependent of the employed storage policy and routing strategy, are neglected.

4. Computational study

We examine the DPA in a picker-to-parts warehouse that is adopted in [6] and a real data where the demands of 800 products over 52 weeks (periods) are recorded. The warehouse has 20 two-sided aisles and 20 storage locations per aisle side. There is only one depot that is located on one side of the first aisle in the warehouse. The distance between two adjacent aisles and storage locations are 4m and 1m.

Let $M(\cdot)$ be the dedicated demand-based storage policy, which assigns higher demanded products to storage locations closer to the depot and initial SLA $A_0 = M(O_{1,2,...,52})$. $L(A, O)$ is calculated by assuming that every product in O is picked one by one under the given A.

To show the effectiveness of DPA, we also examine the performances of intuitive decisions, i.e., reassigning every week, month or quarter. The reassignment cost C is evaluated by the incurred travel distance and its value varies for different situations. We examine three cases of C, i.e., low (10000m), medium (20000m) and high (30000m). For each case, the order-picking cost (O), reassignment cost (R) and the total costs (T) of the aforementioned methods are calculated and displayed in Table 1.

Computational results show that DPA outperforms the intuitive decisions in total operation cost. The running time of DPA is less than 1s on a Windows platform with Intel i7-3770 CPU. It is benefic to reassign SLA in every week if C is small. When C is big enough, however, remaining the initial SLA is a good choice. It should be noted that DPA is more efficient when product demands significantly fluctuate over time.

Table 1. Computational results (unit: m).

	C = 10000			C = 20000			C = 30000		
Method	O	R	T	O	R	T	O	R	T
DPA	16134234	510000	**16644234**	16710312	140000	**16850312**	16867592	0	**16867592**
Week[1]	16124402	520000	16644402	16124402	1040000	17164402	16124402	1560000	17684402
Month[2]	16621094	130000	16751094	16621094	260000	16881094	16621094	390000	17011094
Quarter[3]	16761586	50000	16811586	16761586	100000	16861586	16761586	150000	16911586

[1] Reassign every week.
[2] Reassign every 4 weeks.
[3] Reassign every 12 weeks.

5. Conclusion

It is usually benefic for online retailers to provide fast product delivery service (PDS) which ensures that customers can receive their ordered products in short time. The PDS improvement requires warehouses or distribution centers to fulfill customer orders within tight time windows, in which order-picking is the most time-consuming and labor-intensive activity. In literature, various storage policies are proposed to improve the order-picking efficiency, but most of them disregard the fact that many products' demands fluctuate over time. Moreover, in a long planning horizon, warehouse managers may concern that when to update the warehouse storage location assignment to meet the demand trends. We refer to the problem as storage location assignment planning problem (SLAPP). In this paper, we formulate SLAPP as an mathematical programming and develop a dynamic programming approach (DPA) to solve it. The experimental results indicate the following managerial implications:

- Warehouse storage plan that considers product demand fluctuation, reduces the order-picking effort in terms of total travel distance, and in consequence is benefic for online retailers to implement better PDSs.
- Storage reassignment cost C makes a great difference on storage planning: The smaller (larger) C is, the more (less) frequently storage assignment should be updated.
- With the proposed DPA, we can evaluate the demand fluctuation and find a good balance between reassignment cost and order-picking cost. It should be noted that DPA is independent of order-picking systems, which means we can apply DPA for warehouse employing various order-picking systems.

In this study, we evaluate the reassignment cost as given travel distance and assume that it is constant. In practice, however, the reassignment cost is usually dependent on the warehouse shape, current and target storage assignment, order-picking system, etc. Therefore, we will relax the mentioned assumption and develop efficient approach in the future research.

Acknowledgements

The work is supported by the National Natural Science Foundation of China with grant No.71571006. The authors thank the referees for their encouraging comments and suggestions for improving the quality of the paper.

References

[1] Boysen, N., De Koster, R., Weidinger, F., 2018. Warehousing in the e-commerce era: A survey. European Journal of Operational Research In press.

[2] Dijkstra, A.S., Roodbergen, K.J., 2017. Exact route-length formulas and a storage location assignment heuristic for picker-to-parts warehouses. Transportation Research Part E Logistics & Transportation Review 102, 38–59.

[3] Gils, T.v., Ramaekers, K., Caris, A., De Koster, R.B., 2018. Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review. European Journal of Operational Research 267, 1–15.

[4] Petersen, C.G., Schmenner, R.W., 1999. An evaluation of routing and volume - based storage policies in an order picking operation. Decision Sciences 30, 481–501.

[5] Richards, G., 2014. Warehouse Management: A Complete Guide to Improving Efficiency and Minimizing Costs in the Modern Warehouse. Kogan Page, London.

[6] Xiao, J., Zheng, L., 2010. A correlated storage location assignment problem in a single-block-multi-aisles warehouse considering BOM information. International Journal of Production Research 48, 1321–1338.

[7] Yu, Y., De Koster, R.B.M., Guo, X., 2015. Class-based storage with a finite number of items: Using more classes is not always better. Production and Operations Management 24, 1235–1247.

[8] Zhang, R., Wang, M., Pan, X., 2019. New model of the storage location assignment problem considering demand correlation pattern. Computers & Industrial Engineering 129, 210–219.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.