Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Computers & Operations Research 31 (2004) 623–636
computers & operations research
www.elsevier.com/locate/dsw

Product location, allocation and server home base location for an order picking line with multiple servers

Elizabeth Jewkes[a,*], Chulung Lee[b], Ray Vickson[a]

[a] *Department of Management Sciences, University of Waterloo, Waterloo, Ont., Canada N2L 3G1*
[b] *Department of Industrial and Systems Engineering, National University of Singapore, Singapore 119260, Singapore*

Abstract

In this paper, we are interested in several interrelated control issues for a 'pick to order' (or, 'strict' order picking) picking line which stores $N = nk$ types of products in n bins, each with k shelves. To fill each order, a container is transported past the various locations containing products, and the appropriate quantity of each product is removed from its respective storage location and put into the order container using an 'out and back' picking strategy. Each of several pickers is assigned a set or 'zone' of products. We are interested in the concurrent problems of: (1) product location, (2) picker home base location, and (3) allocating products to each picker so that the expected order cycle time is minimized. We provide easily implemented algorithms to solve these problems and are able to show that the results apply for several alternate picking strategies. For fixed product locations, we develop an efficient dynamic programming algorithm which determines the optimal product allocation and server locations.

Keywords: Warehousing; Order picking; Product location; Server location; Zoning

1. Introduction

In recent years, an increase in time-based competition and the growth of e-commerce have put pressure on warehouse managers to streamline warehouse operations (Shamlaty [1]). Trade journals are full of success stories illustrating their solutions to warehouse design and control problems (e.g. Maloney [2], Ruriani [3]). These problems include warehouse layout, stock profiling, picking strategies and how to set stock location. The focus of this paper is on order picking. Coyle et al. [4] point out that up to 65% of the operating costs of a warehouse can be attributed to order picking.

* Corresponding author.
E-mail address: emjewkes@engmail.uwaterloo.ca (E. Jewkes).

0305-0548/04/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0305-0548(03)00035-2

In this paper, we are interested in several interrelated control issues for a 'pick to order' (or, 'strict') order picking line. A 'pick to order' strategy is one where a picker works on one order at a time—a commonly used approach for small- to medium-sized items such as health and beauty, household, office or food products where the items can be stored in relatively small and accessible pick locations along the picking line (Maloney [5]). To fill each order, a container is transported past the various locations containing products and the appropriate quantity of each product is removed from its respective storage location and put into the order container. With such lines, it is common to utilize several order pickers, each of whom is assigned a set or 'zone' of products and a 'pick and pass' strategy is used to move the order container down the line (see, e.g. Maloney [2]). Material handling equipment such as rolling carts, flow racks, carousels and conveyors are examples of flexible, cost-effective alternatives suitable for such environments. Light-directed picking, bar coding, radio frequency (RF) handheld devices and computer display units are examples of equipment used to assist pickers in the process of filling orders accurately and quickly.

In their review paper, Johnson and Brandeau [6] summarize a variety of design and control issues for material storage and retrieval systems. We assume here that major design issues such as the size of the picking line and the number of pickers are fixed, as is the design of the material handling system. Our focus is on a set of control issues for a 'pick to order' line, as described earlier, in which each picker operates from a fixed home base and uses an 'out and back' picking strategy (see e.g. Wilson [7]) to pick items from its zone before passing the order container to the next picker. The home base could be, for example, a scale and labelling system (for meats, or customized product labels) or a product scanning station/display used for inventory control or order management. We are interested in the concurrent issues of (1) product location, (2) allocating products to each picker, (i.e. zoning the line) and (3) picker home base location. Our objective is to maximize the expected number of orders that can be completed per unit time, or, equivalently, to minimize the expected order cycle time.

While there is a large literature on automated storage and retrieval systems (ASRS) and warehouse design and control, (see e.g. Muralidharan et al. [8], Thonemann and Brandeau [9], Vickson and Lu [10]), we have seen little research on these issues when considered jointly.

This paper is organized as follows: in the next section, we provide a literature review. Section 3 provides the problem definition, and Section 4 presents the optimal solution to the problem of product location, allocating products to each picker and finding the optimal home base locations. The optimality of greedy bin packing for long and low two-dimensional picking lines is also presented. Section 5 gives an efficient dynamic programming solution to the problem of finding the optimal home base locations and product allocations when the product locations are already fixed. Our conclusions can be found in Section 6.

2. Literature review

We are interested here in the concurrent issues of (1) product location, (2) picker home base location, and (3) allocating products to each picker for an 'out and back' picking line so that the expected order cycle time is minimized. The issue of product location for rectangular or linear storage racks has received much attention. Hausman et al. [11], Graves et al. [12] and Jarvis and McDowell [13] among others examine a number of product location strategies for a single server operating from a fixed I/O point. Many other authors have extended the early work of Hausman, Schwarz and Graves to handle interleaving of storage/retrieval requests, and other travel time metrics (see a review in Johnson and Brandeau [6]). Stern [14], Fujimoto [15] and Vickson and Fujimoto [16] have addressed storage schemes for non-anticipatory server location for carousel storage rack. In a non-anticipatory scheme, the carousel remains stationary at the picking station until the next pick request occurs. All of this work indicates the appeal of sorting the products in descending demand order, and then placing them in the storage locations in sequence or in an alternating organ pipe arrangement (OPA). In the context of computer disk systems, similar file location schemes have been addressed by, for example, Yue and Wong [17], and Grossman and Silverman [18].

‘Dwell point’ research has been focussed on the optimal location for a storage/retrieval machine to wait in anticipation of the next storage or retrieval request. In this research, the dwell point is the location of the storage/retrieval machine when it becomes idle, distinct from its I/O location. Bozer and White [19], Egbelu [20], Egbelu and Wu [21,22] and Peters et al. [23] look at a variety of such issues for systems in which product location and the I/O point(s) are assumed fixed or random. King [24], Vickson et al. [25], Gerchak and Lu [26] have addressed optimal anticipatory read/write head location in the context of computer disk systems. Anderson and Fonenot [27] deal with optimal home base locations for multiple patrol cars, which respond to independent and identically distributed requests from a set of locations on a line.

Several authors have looked at product location jointly with other decision variables in warehouse design and control: Muralidharan et al. [8] examine heuristics in which an ASRS shuffles product locations when otherwise idle to minimize waiting time and service time. Both Thonemann and Brandeau [9] and Wilson [7] consider product location jointly with the use of a specific inventory policy in order to minimize order picking and inventory costs. Foulds and Hamacher [28] consider both optimal bin location and sequencing in printed circuit board assembly. Vickson and Lu [10] find jointly optimal product and server locations for a single server in linear and carousel storage racks. In this paper, we extend the work of Vickson and Lu to multiple servers and work with slightly different product demand definitions.

Despite references to the use of zones in picking lines (e.g. Maloney [2], www.lighteningpick.com, Frazelle and Apple [29]), there appears to be little analytical research on how to construct picker zones. Bassan et al. [30] examine two shelf configurations for a homogeneous and zoned warehouse and compare costs associated with material handling, warehouse perimeter and area to determine the annual cost-minimizing configuration. Bartholdi and Eisenstein [31] look at a “Toyota Sewn Products Management System” (TSS) production system with n servers working among m stations where $n \leqslant m$. They find that if the servers are sequenced from slowest to fastest, then a natural partition of work will emerge and the production rate will be simultaneously maximized. A number of other authors have looked at the issue of partitioning schemes for AGVS networks (see Johnson and Brandeau [6]). Such work typically looks at issues of throughput capacity or workload balancing both within and between loops so that total material handling efforts are minimized. Anderson and Fontenot [27] deal with the issue of optimal home base location for multiple servers (patrol cars), which respond to independent and identically distributed demands from locations on a line. Peterson [32] conducted a simulation study comparing several picking strategies for zoned mail-order companies. The zones were constructed so that each contained an equal proportion of expected demand, and within a zone, high demand SKUs were stored closest to the end of an aisle where the conveyor was located. As we shall see, these simulation results are consistent with our analytical work.

In summary, previous studies have either considered only the case of a single server or have not considered product and server location simultaneously with allocating products to multiple servers.

3. Problem description

Consider a picking line using a carton or pallet flow rack system, which stores N products. Storage locations on the line are arranged in n stacks of shelves, spaced equally on the picking line. Each stack contains $k \geqslant 1$ shelves of equal size so that there are $kn = N$ storage locations. Each product consumes a single storage location. The picking line is assumed to be long and low, such that it would be reasonable for a picker to reach all storage locations within a vertical shelf (henceforth called a *bin*) in the same amount of time.

Customer orders, which consist of demands for one or more products, are filled by $s \geqslant 1$ pickers (or servers) in sequence by using a 'pick and pass' strategy. Partially filled orders are transported down the line, from server to server by a material handling device such as a roller conveyor (refer to Fig. 1 for a conceptual diagram). Each server operates from a 'home base', and is responsible for picking products from a specified set of bins on the line, called a 'zone'. When an order container arrives at the servers' home base, (s)he begins picking items in the order from the set of products within his or her zone.

Successive orders are assumed to be independent, and demand for individual products within an order are assumed to be independent of one another (i.e. we ignore issues such as product commonality). The number of items of product p in a random order, N_p, is a random variable with a probability mass function $h_p(\cdot)$, $p = 1, \ldots, N$. Then,

$$\begin{aligned}\lambda_p &= Pr(\text{an order contains demand for one or more items of product } p)\\ &= 1 - h_p(0), \quad p = 1, \ldots, N. \end{aligned} \tag{1}$$

In terms of travel time, we do not distinguish between the k products stored in each vertical shelf and (initially) assume that a picker is able to retrieve all items in an order from a single bin in one trip. The aggregate probability that an order will require the picker to travel to bin j will then be

$$\begin{aligned}P_j &= Pr(\text{an order contains demand for one or more items from bin } j)\\ &= 1 - \prod_{p \in \text{bin} j} h_p(0), \quad j = 1, \ldots, n. \end{aligned} \tag{2}$$

In the sequel, reference to product j is taken to mean the *set* of products stored in bin j. We will deal later with alternate definitions for the $\{P_j\}$ when the picker makes a separate trip for

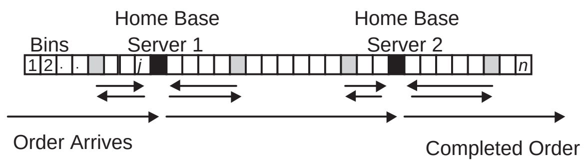

Fig. 1. Illustration of picking line.

each type of product within a bin, or when the picker retrieves each item in an order individually. We also defer the problem of how to optimally form the N product types into n groups of k and assume that the aggregate product probabilities $\{P_j\}$ are given and are sorted in decreasing order $P_1 \geqslant P_2 \geqslant \cdots \geqslant P_n > 0$.

To pick products from a bin, a server moves from its home base to the storage bin, the appropriate items are removed, and then the server returns to its home base, completing any item processing prior to placing the item in the order container. Examples of such processing are weighing (e.g. meats), inspecting, tagging or swiping the product barcode for inventory control. Once the server has picked all products in the order from those within his/her zone, the container is carried by the material handling system to the next server on the line. The order is complete when it has passed through each of the servers in sequence.

Server i travels at a constant velocity, v_i $i = 1,\ldots,s$ bins/unit time, and we assume that server i is the ith in the sequence of servers on the line. The distance between bin m and bin l is denoted by $d(m,l) = |m - l|$ so that the travel time for server i to get from bin m to bin l is $(1/v_i)d(m,l)$.

The time required to pick items from the bins and to process the items is not included in the expected order cycle time, as these times are independent of product location and assignment of products to servers. We also assume that, once a server is finished picking the products for its zone, the material handling system independently carries the container to the next zone. Since it is reasonable to assume this inter-zone travel time to be constant for each order, we also omit the time required for the order to move between servers from the cycle time. Our objective is to determine an optimal arrangement of the $\{P_j\}$, which products to allocate to each server and where to locate the home base for each server in order to minimize the expected cycle time for a random order.

We denote by σ any assignment of the probabilities $\{P_j\}$ to bins. Given an arbitrary σ, we define $P_{\sigma(j)}$ as the probability associated with bin j. The set of home base locations (bins) for the servers is denoted by $\omega = \{\omega_1, \omega_2, \ldots, \omega_s\}$ and finally $\pi = \{\pi_1, \pi_2, \ldots, \pi_s\}$ is the set of products (and associated bins) allocated to each server.

With these definitions and assumptions, we can write the expected cycle time associated with an arbitrary assignment of $\{P_j\}$ to bins, server home base locations, and allocation of products to servers, denoted by (σ, ω, π), as

$$C(\sigma,\omega,\pi) = 2\sum_{i=1}^{s} \frac{1}{v_i} \sum_{j\in\pi_i} P_{\sigma(j)}\, d(\omega_i, j). \tag{3}$$

Note that (3) includes the server's travel time from the home base location to each bin being picked *and* the return travel time. Without loss of generality, we henceforth ignore the return travel time and omit the factor of 2 in the expected cycle time.

4. The optimal product assignment, product allocation and home base locations

In this section, we assume that product location, server home base location and the allocation of products to servers are to be jointly determined in order to minimize the expected cycle time of a random order. We begin by pointing out several characteristics of the coefficients of $P_{\sigma(j)}$ in Eq. (3). The maximum number of bins that can be potentially allocated to server i is n. If server i has its home base at bin 1 (or n), travel times to bins it serves are $0 = 0/v_i, 1/v_i, 2/v_i, \ldots, (n-1)/v_i$, in non-decreasing travel time order. If server i has its home base at bin 2 (or $n-1$), travel times in non-decreasing order are $0=0/v_i, 1/v_i, 1/v_i, 2/v_i, \ldots, (n-2)/v_i$. Taking all possible locations of the home base for server i into account, the coefficients of the $P_{\sigma(j)}$ in Eq. (3) can be characterized in the following way: The term 0 is always present, the terms d/v_i are present twice for $d=1,\ldots,\lceil n/2\rceil-1$, and the terms d/v_i are present once for $d=\lceil n/2\rceil,\ldots,n-1$, where $\lceil x\rceil$ is the smallest integer greater than or equal to x. Therefore, the list of potential travel times for server i is

$$f_i = \frac{1}{v_i}(0,1,1,\ldots,\lceil n/2\rceil-1,\lceil n/2\rceil-1,\lceil n/2\rceil,\lceil n/2\rceil+1,\ldots,n-1),$$

a vector of dimensionality $n+\lceil n/2\rceil-1$. Let $K=s\times(n+\lceil n/2\rceil-1)$ and $F=(f_1,f_2,\ldots,f_s)$ be the K-vector obtained by concatenating the vectors f_i, and $F^*=(F_i^*,\ldots,F_K^*)$ be the vector obtained by sorting the components of F into ascending order. If ties occur between components of F^* which correspond to server i and j, they are broken by listing the component corresponding to server i first if $i<j$ or by perturbing v_i by a small positive quantity, ε. Note that components $1,\ldots,s$ of F^* are equal to zero, components $s+1$ and $s+2$ are equal to $1/\max\{v_i\}$, and so forth. Finally, let $Q^*=(P_1,P_2,\ldots,P_n,0,0,\ldots,0)$ be the K-vector obtained by appending zeroes to the list of P_j. Now, for any policy (π,ω,σ) we have

$$C(\sigma,\omega,\pi)=\sum_{j=1}^{K} Q_j F_j^* \tag{4}$$

for some $Q=$ permutation of Q^*. This holds for any policy in which the home base of a server is at one of its assigned bins.

In general, every policy (π,ω,σ) corresponds to a permutation of Q^* in the sense of Eq. (4), but we do not claim the converse. However, the optimal permutation Q (which is just Q^* itself) does correspond to a policy $(\pi^*,\omega^*,\sigma^*)$ which is thus an optimal policy.

Example 1. Consider the following (non-optimal) policy for the case $n=10$ and $s=2$: (i) assign $\{P_j\}$ to bins $1,2,\ldots,10$ in the order $P_8P_4P_5P_7P_1P_9P_2P_3P_{10}P_6$; (ii) assign server 1 to bins 1, 5, 6 and 7 with home base at bin 1; and (iii) assign server 2 to bins 2, 3, 4, 8, 9 and 10 with home base at bin 9. For this policy the cycle time is

$$C=(1/v_1)(0P_8+4P_1+5P_9+6P_2)+(1/v_2)(0P_{10}+1P_6+1P_3+5P_7+6P_5+7P_4).$$

The 14 vectors of potential travel times are $f_1=(1/v_1)h$ and $f_2=(1/v_2)h$ where $h=(0,1,1,2,2,3,3,4,4,5,6,7,8,9)$. If v_1 and v_2 are 1 and 1.5 bins/unit time, respectively, then the sorted 28-vector

$$F^*=\left(0,0,\frac{2}{3},\frac{2}{3},1,1,\frac{4}{3},\frac{4}{3},2,2,2,2,\frac{8}{3},\frac{8}{3},\ldots,8,8,9\right)$$

$$\equiv\left(\frac{0}{v_1},\frac{0}{v_2},\frac{1}{v_2},\frac{1}{v_2},\frac{1}{v_1},\frac{1}{v_1},\frac{2}{v_2},\frac{2}{v_2},\ldots,\frac{8}{v_2},\frac{9}{v_2}\right).$$

Obviously, the expression for the cycle time can be written in the form of Eq. (4) for an appropriate permutation of $Q^*=(P_1,\ldots,P_{10},0,\ldots,0)$.

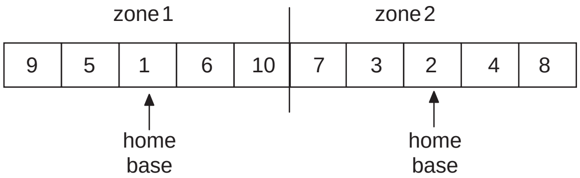

Fig. 2. Optimal zones, home bases and product-bin allocations in Example 1.

Determining the permutation Q which minimizes $\sum_i F_i^* Q_i$ is simple on the basis of the following well-known result (see, eg., Marshall and Olkin [33]):

Lemma 1. *If* $a=(a_1,a_2,\ldots,a_K)$ *and* $b=(b_1,b_2,\ldots,b_K)$ *are non-negative and* $0 \leqslant a_1 \leqslant a_2 \leqslant \cdots \leqslant a_K$, *the sum* $\sum_{j=1}^{K} a_j b_{\sigma(j)}$ *is minimized over permutations* σ *of* b *by the ordering* $b_{\sigma(1)} \geqslant b_{\sigma(2)} \geqslant \cdots \geqslant b_{\sigma(K)}$ (*opposite ordering*).

This result implies that Q^* is the optimal permutation, so only the first n elements of F^* need to be retained. The vector F^* may thus be truncated to its first n components, after which the optimal cycle time objective is simply: $C(\sigma^*,\omega^*,\pi^*)=\sum_{j=1}^{n} F_j^* P_j$. In the sequel, we let $g^*=(g_1^*,g_2^*,\ldots,g_n^*)$ denote the vector F^* truncated to its first n components.

The optimal policy is obtained from g^* as follows: (i) each server serves at least one bin, because all s of the initial components of g^* are assigned to some P_j; (ii) the s highest demands $P_1,\ldots,P_s$ are assigned to different servers; (iii) each server's home base is at the highest assigned P_j, $j=1,\ldots,s$; (iv) the other bins assigned to server i, and their assigned P_j, are obtained by noting that if $g_r^*=d/v_i$, then P_r is assigned to server i at a distance of d bins from the home base. Either both terms of the form d/v_i will be assigned to server i, or the nth component of g^* is reached. Therefore, the optimal zone for server i will include both bins at distance d from ω_i unless the zone is at the start or end of the line, in which case it is possible to have one of the d-distance bins included but not the other. Furthermore, if a d-distance bin is included in the zone of server i, then so are the pairs of d'-distance bins for all integers $d'<d$. This means that a server's optimal zone is a set of contiguous bins with no gaps, and the zones of different servers are non-overlapping. Also, the home base is at the centre of the zone.

Example 1 (continued). The optimal cycle time for Example 1 is $\sum_{j=1}^{10} g_j^* P_j$ where

$$g^*=\left(0,0,\frac{2}{3},\frac{2}{3},1,1,\frac{4}{3},\frac{4}{3},2,2\right)=\left(\frac{0}{v_1},\frac{0}{v_2},\frac{1}{v_2},\frac{1}{v_2},\frac{1}{v_1},\frac{1}{v_1},\frac{2}{v_2},\frac{2}{v_2},\frac{2}{v_1},\frac{2}{v_1}\right)$$

is F^* truncated to its first $n=10$ components. The corresponding optimal policy is as shown in Fig. 2. The expected cycle time is 0.43 time units.

4.1. Optimal product groupings

We now turn to the issue of how to construct n groups of k products to form optimal bin probabilities $\{P_j\}$. As in Section 3, we denote by $h_p(\cdot)$ the probability mass function of the number of items of product $p=1,2,\ldots,N$ in a random order, and by $\lambda_p=1-h_p(0)$ the probability that such an order includes one or more units of product p. Without loss of generality, we sort the λ_p in decreasing order $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \lambda_N$ which implies $h_1(0) \leqslant h_2(0) \leqslant \cdots \leqslant h_N(0)$. If k products $p \in G_j$ are included in the jth group, the probability of an order containing one or more products in G_j is $P_j = 1 - \prod_{p \in G_j} h_p(0)$. Below, we show that the following 'greedy' scheme produces optimal bin probabilities $\{P_j\}$.

4.1.1. Greedy product allocation

Aggregate the N products into n groups of k, starting with product 1 and proceeding sequentially to product N in ascending $h_p(0)$ order. Thus, the optimal first group is $G_1^* = \{1, 2, \ldots, k\}$, the optimal second group is $G_2^* = \{k+1, k+2, \ldots, 2k\}$, and so forth.

Our results in the previous section imply that for given $P_1 \geqslant P_2 \geqslant \cdots P_n$, the optimal expected cycle time is $C = \sum_{j=1}^{n} P_j g_j^*$. Optimality of the greedy probabilities $P_j^* = 1 - \prod_{p=(j-1)k+1}^{jk} h_p(0)$ is obtained by sequential application of the following result.

Lemma 2. *Suppose that $P_j \geqslant P_{j+1}$ for $j < n$, but the groups G_j contain products a and b such that $h_a(0) > h_b(0)$ for some $a \in G_r, b \in G_s$ with $r < s$. Then interchanging products a and b does not increase the expected cycle time, and decreases it strictly if $g_s^* > 0$.*

Proof. $P_r = 1 - \prod_{p \in G_r} h_p(0) \equiv 1 - h_a(0) R_r$ and $P_s = 1 - \prod_{p \in G_s} h_p(0) \equiv 1 - h_b(0) R_s$. Since $P_r \geqslant P_s$ and $h_a(0) > h_b(0)$, we have $R_r < R_s$. After interchanging products a and b the new bin probabilities are $P'_j = P_j$ for $j \neq r, s$, $P'_r = 1 - h_b(0) R_r$ and $P'_s = 1 - h_a(0) R_s$.

Letting C and C' denote the cycle times before and after the interchange, we have

$$\begin{aligned} C - C' &= [1 - h_a(0)R_r]g_r^* + [1 - h_b(0)R_s]g_s^* - [1 - h_b(0)R_r]g_r^* - [1 - h_a(0)R_s]g_s^* \\ &= [h_a(0) - h_b(0)](R_s g_s^* - R_r g_r^*) \geqslant 0, \end{aligned}$$

using previous inequalities and the fact that $g_s^* \geqslant g_r^*$. Furthermore, $C - C' > 0$ if $g_s^* > 0$. Since $P'_r > P_r$ and $P'_s < P_s$, the $\{P'_j\}$ after the interchange might need to be re-sorted to put them in descending order again. However, this would merely decrease the cycle time even more. Sequential application of Lemma 2 implies that an optimal grouping is obtained by making P_1 as large as possible, then making P_2 as large as possible among the remaining products, and so forth. □

Example 2. Consider the $N = 21$ product example with $n = 7$ bins, $k = 3$ shelves per bin and $s = 2$ servers and product demand rates as shown in Table 1. The optimal product groupings and the resulting $\{P_j\}$ are shown with the result that servers 1 and 2 are located at bins 2 and 6, respectively. The optimal cycle time is 2.8796 time units/order.

Remark. (1) The optimal policy is not unique, even in the absence of ties when making choices. The largest bin probabilities $P_1, \ldots, P_s$ all appear with coefficients of zero in the objective, so the first $s \times k$ products should be assigned to the first s groups. However, it is not strictly *necessary* to have P_1 as large as possible, then P_2 as large as possible after that, etc. Other allocations would be as good, since they correspond to zero-improvement interchanges in the proof of Lemma 2.

(2) When assigning two adjacent values P_j and P_{j+1} to bins on opposite sides of a server's base, either orientation may be selected.

Table 1
Example of greedy bin packing

Product (p)			$h_p(0)$
Sorted demand data			
1			0.0097
2			0.0268
3			0.0388
4			0.0464
5			0.1036
6			0.1712
7			0.2030
8			0.2283
9			0.2836
10			0.3184
11			0.3281
12			0.6883
13			0.7301
14			0.7381
15			0.7666
16			0.8452
17			0.9423
18			0.9639
19			0.9642
20			0.9799
21			0.9814
j	Products	P_j	g_j^*
Optimal product grouping			
1	1,2,3	0.99999	0
2	4,5,6	0.99918	0
3	7,8,9	0.98686	1
4	10,11,12	0.92810	1
5	13,14,15	0.58687	1
6	16,17,18	0.23230	1
7	19,20,21	0.07272	2
Bin (j)		Products	
Optimal product and server locations			
1		7,8,9	
2[a]		**1,2,3**	
3		10,11,12	
4		19,20,21	
5		13,14,15	
6[a]		**4,5,6**	
7		16,17,18	

$C = 2.8796$.
[a]Home base location.

(3) In Section 3, we assumed that all items in an order within the same bin are retrieved by the picker in one trip. Two alternate picking strategies that give rise to slightly different definitions for $\{P_j\}$, but which do not change our basic results are the following:

(i) If all items of the same *type* (i.e. in the same shelf) are retrieved in one trip, several trips to the same *bin* may be required, then

$$P_j = \sum_{p \in G_j} [1 - h_p(0)] = \sum_{p \in G_j} \lambda_p, \quad j = 1, \ldots, n. \tag{5}$$

(ii) If each *item* is retrieved in a single trip, the *expected* cycle time can be calculated using

$$P_j = \sum_{p \in G_j} E(N_p), \quad j = 1, \ldots, n, \tag{6}$$

where $E(N_p)$ is the expected number of items of type $p, p = 1, \ldots, N$ in a random order.

For both alternatives to our original definition of $\{P_j\}$, the results for product and server location remain unchanged. The proof that greedy bin packing is optimal carries through with minor changes. In scenario (ii) the products must be sorted into descending $E(N_p)$ order instead of descending λ_p order.

5. A dynamic programming (DP) algorithm for fixed product locations

In this section, we assume that the product locations are given, and denote by P_j the probability that a random order will contain one or more items from bin $j, j = 1, \ldots, n$, as in Eq. (2), though we note that our results also apply for the alternate definitions in Eqs. (5) and (6).

With given product locations, the product allocation and server home base locations which minimize the expected cycle time is the s-median of $\{P_j\}$. While the general s-median problem is known to be NP-hard (Kariv and Hakimi [34]), for graphs with a tree structure (as in our case), Kariv and Hakimi were able to develop a $O(n^2 s^2)$ polynomial time algorithm to solve the s-median problem with n nodes. We present below an $O(n(n-s)^2)$ DP algorithm which is easily implemented and more tailored to our problem than the Kariv and Hakimi algorithm.

The optimization problem is formulated as an s-stage deterministic DP problem where in stage i $(i = 1, \ldots, s)$, we jointly determine the products (bins) allocated to server i and the home base location for server i. This stage-wise formulation can deal with both the case of servers with equal velocities, or where the servers are of unequal velocity, but are assigned to the line in a specific sequence. In other words, the sequence of server zones is specified, but not the actual zone boundaries or home base locations.

For the DP formulation, the optimal-value function, which we denote by $S(i, l)$, is the minimum total cycle time when the first l bins are picked by the first i servers. The forward-recursion relationship for computing the optimal-value function can be written as follows:

$$S(i, l) = \min_{i-1 \leqslant q \leqslant l-1} \left\{ S(i-1, q) + \frac{1}{v_i} M(q+1, l) \right\}, \quad i = 2, \ldots, s, \;\; l = i, \ldots, n-s+i, \tag{7}$$

where

$$M(j,l)=\sum_{k=j}^{l}P_k d(k,\omega^*(j,l))=\min_{j\leqslant q\leqslant l}\sum_{k=j}^{l}P_k|k-q| \tag{8}$$

and $\omega^*(j,l)$ denotes the median of (contiguous) bins $j,\dots,l$. The boundary conditions are

$$S(1,q)=(1/v_1)M(1,q), \quad q=1,\dots,n-s. \tag{9}$$

The optimal cycle time $S(s,n)$ can be computed recursively with (7)–(9), and the optimal allocation of bins to servers and server home base locations can be obtained by backtracking through the optimal policies in each of the previous stages.

To increase the computational speed of our algorithm, we can reduce the number of states considered in each stage by removing those that have a cycle time greater than an upper bound on the cycle time. Lemma 3 gives such an upper bound:

Lemma 3. *An upper bound for the optimal* (*minimum*) *travel time is*

$$UB=\frac{1}{2}\sum_{j=1}^{n}P_j\left\lceil\frac{n}{\sum_{i=1}^{s}v_i}\right\rceil. \tag{10}$$

Proof. A feasible solution can be obtained by first dividing the linear rack into s subsets such that the travel times for servers to travel the entire length of their zones are equal, and placing a home base at the centre bin in each zone. In this solution, the travel time from a home base to the farthest bin in its zone will not be greater than $(1/2)\lceil n/\sum_{i=1}^{s}v_i\rceil$. The upper bound is easily obtained by assuming that travel to each bin has this worst case travel time. □

Example 3. The DP algorithm above was applied to the 12-product, storage rack with fixed product locations with $\{P_j\}=\{0.2,0.8,0.4,0.7,0.6,0.3,0.3,0.2,0.4,0.6,0.4,0.5\}$. Products in the storage rack are picked by two servers where $v_1=1$ bin/unit time and $v_2=2$ bins/unit time. Our objective is to allocate products to servers, and to find the server home base locations to minimize the expected order picking time.

First, Lemma 3 provides an upper bound for the total travel time in the system of $UB=10.8$ time units. We determine the number of bins served by server 1 that achieves the minimum travel time. The boundary conditions are:

$$S(1,1)=0.00 \quad (q=1,\omega_1=1),$$

$$S(1,2)=0.20 \quad (q=2,\omega_1=2),$$

$$S(1,3)=0.60 \quad (q=3,\omega_1=2),$$

$$S(1,4)=1.90 \quad (q=4,\omega_1=3),$$

$$S(1,5)=3.10 \quad (q=5,\omega_1=3),$$

$$S(1,6)=3.80 \quad (q=6,\omega_1=4),$$

$$S(1,7)=4.70 \quad (q=7,\omega_1=4),$$

$$S(1,8) = 5.50 \quad (q = 8, \omega_1 = 4),$$

$$S(1,9) = 7.50 \quad (q = 9, \omega_1 = 4),$$

$$S(1,10) = 10.80 \quad (q = 10, \omega_1 = 5),$$

$$S(1,11) = 13.20 \quad (q = 11, \omega_1 = 5),$$

$S(1,11) > UB$, thus $S(1,11)$ and $S(1,12)$ will not be considered. The computations for the second server are as follows:

$$S(2,l) = \min_{1 \leqslant q \leqslant l-1} \left\{ S(1,q) + \frac{1}{v_2} M(q+1,l) \right\}, \quad l = 2,\ldots,12,$$

$$S(2,12) = 5.25 = \min \begin{cases} 7.85 & (q = 1, \omega_2 = 6), \\ 6.25 & (q = 2, \omega_2 = 7), \\ 5.75 & (q = 3, \omega_2 = 8), \\ 5.40 & (q = 4, \omega_2 = 9), \\ 5.25 & (q = 5, \omega_2 = 10), \\ 5.35 & (q = 6, \omega_2 = 10), \\ 5.80 & (q = 7, \omega_2 = 10), \\ 6.40 & (q = 8, \omega_2 = 10), \\ 8.05 & (q = 9, \omega_2 = 11), \\ 11.0 & (q = 10, \omega_2 = 12), \\ 13.20 & (q = 11, \omega_2 = 12). \end{cases}$$

The optimal cycle time is 5.25 time units. In the optimal policy, server 1 has its home base at bin 3 and is allocated bins 1–5. Server 2 has its home base at bin 10 and serves bins 6–12 (i.e. $\omega_1^* = 3$, $\pi_1^* = \{1,2,3,4,5\}$, $\omega_2^* = 10$, and $\pi_2^* = \{6,7,8,9,10,11,12\}$).

The computational effort of the DP algorithm can be reduced by computing in advance the total travel times for every set of bins that can be served by a single server. For each of n bins, at most $n - s + 1$ bins will be handled by the same server. Given a set of bins served by a single server, $O(n - s)$ operations are necessary to locate an optimal home base by Goldman's [35] algorithm. These pre-processing steps require $O(n(n - s)^2)$ operations and can be reused in each state of the DP algorithm to compute the optimal value functions. In the algorithm, no more than $n - s + 1$ bins must be considered to compute an optimal value function in a particular state. In addition, there are at most $n - s + 1$ states in each stage. Thus, the execution of the DP algorithm requires $O(s(n - s)^2)$ operations. Since the pre-processing step, which is executed once, has a higher computational complexity, $O(n(n - s)^2)$, it determines the computational effort of the proposed algorithm.

6. Conclusions

In this paper, we were interested in several interrelated control issues for a multiple-picker order picking line which stores $N = nk$ types of products in n bins, each with k shelves. Pickers operate from a home base location and use 'out and back' picking strategy to fill orders. We determined the optimal policy for the concurrent problems of (1) product location, (2) picker home base location, and (3) allocating products to each picker so that the expected order cycle time is minimized. We were able to show that a greedy approach is optimal for grouping products into bins and that our results apply for several alternate 'out and back' picking strategies. For fixed product locations, we developed an efficient dynamic programming algorithm which determines the optimal product allocation and server locations.

The research in this paper can be extended in several ways. First, in this paper, we assumed that the storage racks are long and low, i.e. the vertical travel time is negligible. One extension of our work, suitable for larger systems, is to incorporate vertical travel times into the problem formulation. Another direction of work is to consider picking lines in which the pickers have no home base. The picker fills the order by 'sweeping' from the start of their zone, travelling down the zone only as far as the order requires. In this case, we would be interested in product location and zone formation.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada. The authors would like to thank an anonymous referee for helpful comments.

References

[1] Shamlaty R. This is not your father's warehouse. IIE Solutions 2000; January: 31–36.
[2] Maloney D. The new corner drugstore. Modern Materials Handling 2000; May: 58–64.
[3] Ruriani DC. Small shipments? No tall order. Warehousing Management 2000; January: 27–34.
[4] Coyle JJ, Bardi EJ, Langley CJ. The management of business logistics. St. Paul, MN: West, 1996.
[5] Maloney D. Seeing the light. Modern Materials Handling 2000; August: 49–53.
[6] Johnson ME, Brandeau ML. Stochastic modeling for automated material handling system design and control. Transportation Science 1996;30(4):330–50.
[7] Wilson H. Order quantity, Product popularity and the location of stock in a warehouse. AIIE Transactions 1977;9(3):230–7.
[8] Muralidharan B, Linn R, Pandit R. Shuffling heuristics for the storage location assignment in an AS/RS. International Journal of Production Research 1995;33(6):1161–672.
[9] Thonemann UW, Brandeau M. Note: optimal storage assignment policies for automated storage and retrieval systems with stochastic demands. Management Science 1998;44(1):142–8.
[10] Vickson R, Lu X. Optimal product and server locations in one-dimensional storage racks. European Journal of Operational Research 1998;105:18–28.
[11] Hausman W, Schwarz L, Graves S. Optimal storage assignment in automatic warehousing systems. Management Science 1976;22(6):629–38.
[12] Graves S, Hausman W, Schwartz LB. Storage retrieval interleaving in automatic warehousing systems. Management Science 1977;23:935–45.
[13] Jarvis JM, McDowell ED. Optimal product layout in an order picking warehouse. IIE Transactions 1991;23:93–102.

[14] Stern HL. Parts location and optimal picking rules for a carousel automatic storage and retrieval system. Proceedings of the Seventh International Conference on Automation in Warehousing, San Francisco, 1986. p. 185–92.
[15] Fujimoto A. Models for determining storage strategies for carousel storage and retrieval systems. M.Sc. Thesis, Department of Management Sciences, University of Waterloo, Waterloo, Canada, 1991.
[16] Vickson R, Fujimoto A. Optimal storage locations in a carousel storage and retreival system. Location Science 1996;4:237–45.
[17] Yue PC, Wong CK. On the optimality of the probability ranking scheme in storage applications. Journal of the ACM 1973;20:624–33.
[18] Grossman DD, Silverman HF. Placement of records on a secondary storage device to minimize access time. Journal of the ACM 1973;20:429–38.
[19] Bozer YA, White JA. Travel-time models for automated storage/retrieval systems. IIE Transactions 1984;16:329–38.
[20] Egbelu P. Framework for dynamic positioning of storage/retrieval machines in an automated storage/retrieval system. International Journal of Production Research 1991;29(1):17–37.
[21] Egbelu P, Wu C. A comparison of dwell point rules in an automated storage/retrieval system. International Journal of Production Research 1993;31(11):2515–30.
[22] Egbelu P, Wu C. Relative positioning of a load extractor for a storage carousel. IIE Transactions 1998;30:301–17.
[23] Peters BA, Smith JS, Hale TS. Closed form models for determining the optimal dwell point location in automated storage and retrieval systems. International Journal of Production Research 1996;34(6):1757–71.
[24] King RP. Disk arm movement in anticipation of future requests. ACM Transactions on Computer Systems 1990;8: 214–29.
[25] Vickson R, Gerchak Y, Rotem D. Optimal positioning of read/write heads in mirrored disks. Location Science 1995;3:125–32.
[26] Gerchak Y, Lu X. Optimal anticipatory position of a disk arm of queries of random length and location. Information Systems and Operational Research 1996;34(4):251–62.
[27] Anderson LR, Fontenot RA. Optimal positioning of service units along a coordinate line. Transportation Science 1992;28:346–51.
[28] Foulds LR, Hamacher HW. Optimal bin location and sequencing in printed circuit board assembly. European Journal of Operational Research 1993;66:279–90.
[29] Frazelle EH, Apple JM. Warehouse operations. In: Tompkins JA, Sharp GP, editors. Distribution management handbook. New York: McGraw-Hill, 1996.
[30] Bassan Y, Roll Y, Rosenblatt M. Internal layout design of a warehouse. AIIE Transactions 1980;12(4):317–22.
[31] Bartholdi J, Eisenstein D. A production line that balances itself. Operations Research 1996;44(1):21–34.
[32] Peterson II C. An evaluation of order picking policies for mail order companies. Production and Operations Management 2000;9(4):319–35.
[33] Marshal A, Olkin I. Inequalitites: theory of majorization and its applications. New York: Academic Press, 1979.
[34] Kariv O, Hakimi SL. An algorithmic approach to network location problems II: the p-medians. SIAM Journal of Applied Mathematics 1979;37(3):539–60.
[35] Goldman AJ. Optimal center location in simple networks. Transportation Science 1971;5:212–21.

Further reading

Hakimi SL. Optimum locations of switching centers and the absolute centers and medians of a graph. Operations Research 1964;12:450–9.
Lim, WK, Bartholdi JJ, Platzman LK. Storage schemes for carousel conveyors under real time control. Material Handling Research Centre Technical Report MHRC-TR-85-10, Georgia Institute of Technology, Atlanta, Georgia, 1985.