XXIV. THE CUBE PER ORDER INDEX SLOTTING STRATEGY, HOW BAD CAN IT BE?

Peter C. Schuur
Department of Industrial Engineering and Business Information Systems, University of Twente, Enschede, the Netherlands

Abstract

A well-known and frequently applied policy to assign stock keeping units (SKUs) to (dedicated) storage locations in a warehouse is the Cube per Order Index (COI) slotting strategy. Basically, COI stores an SKU based on how frequently it is picked per unit of stock space required. Fast movers are located close to the Input-Output points. For single command order picking, COI slotting is well-known to minimize order picking travel time. For multi command this is no longer true. An interesting question is: how bad can it be? In this paper we show that there is no limit to this badness. Worst-case behavior of COI is infinitely bad. We construct a worst-case example that proves the following. Given an arbitrary positive integer *p*, there is (i) a warehouse configuration (ii) a set of SKUs (iii) a set of orders for these SKUs such that slotting these SKUs in the warehouse according to COI leads to an order picking travel time which is *p* times larger than the order picking travel time produced by an optimal slotting strategy.

1 Introduction

An important element of warehouse management is the slotting strategy: how to allocate Stock Keeping Units (SKUs) to storage locations. For clarity of exposition, let us confine our discussion to the order picking area of a warehouse. Let us assume this area is man-to-goods. Let there be one Input/Output (I/O) point and let all pick locations be identical except for their distance to the I/O point. Let us assume single order picking, so no batching. And let the order picking device have enough capacity to pick all the items in each order in one tour. Assume furthermore that one aims at minimizing total travel time of the order pickers - given a pre-specified routing policy - during the order picking process. To realize the latter, a dedicated storage slotting strategy is chosen. Dedicated storage means every SKU is uniquely coupled to a specific set of pick locations. Assume that - on the basis of inventory management as well as capacity considerations - it has been decided how many pick locations are to be assigned to every SKU. What remains is the question: given the number of pick locations per SKU, at which specific positions do we allocate them?

In practice, a dedicated storage strategy is often based on the turnover rate or pick frequency of each SKU. This implies that the allocation of SKUs to locations is based on both the popularity of the SKUs and the distances of the locations to the I/O point. A well-known example in literature of this kind of strategy is based on the Cube per Order Index (COI) of an SKU, which is the ratio of the number of locations assigned to this SKU and its pick frequency. It was introduced by Heskett [1][2][3], after which many researchers, Kallina and Lynn [4], Malmborg and Bhaskaran [5][6][7], and Malmborg [8][9] provided further analysis. The COI slotting process comprises the following three steps: (i) the ranking of locations based on their distances to the I/O point in non-decreasing order, (ii) the ranking of SKUs based on their COI in non-decreasing order and (iii) linking the ranked sets of locations and SKUs. In case the number of locations assigned to each SKU is identical, then COI measures the reciprocal of the popularity and the associated slotting strategy simplifies to locating fast movers close to the I/O point.

For *single command* order picking, COI slotting is well-known to minimize order picking travel time. In this case, the order picker begins from the I/O point, performs a retrieval action at a specific location and then returns to the I/O point. However, in reality, multiple items in an order are picked from their respective locations in one tour that begins at the I/O point, visits these locations in a specified sequence and returns to the I/O point. Using COI when multiple SKUs in an order are picked in one tour, means that a single command strategy is implemented for a multi command situation. In such a situation *item oriented* strategies like the COI approach perform worse than *order oriented* strategies as shown in Mantel et al. [10] and De Ruijter et al. [11]. In [10] the average performance of COI on an extensive series of test problems was found to be 11% above optimum. In [11], where COI was compared to order oriented strategies for the practical case of a book distribution center, it turned out that COI improved the current total travel distance by 27%, whereas the best order oriented slotting strategy improved it by 40%.

Interestingly, Heskett - the founding father of COI - shows to be well aware of the working of COI in a multi command situation. Heskett [3] discusses multi command (order-picker system in his terminology) and states: ‘Generally speaking, the cube-per-order index system of location will still offer an accurate general solution because of the desire to place orders, not items, as close as possible to one another. However, in this type of situation more than any other, the stock location planner must lay out items within a zone in a sequence that will provide a logical traffic flow through the zone. He may redesign the shape of order selection zones or create sub-zones to group items associated more closely with each other, not necessarily in relation to some focal point such as the order assembly point.’ In our opinion this clearly points to the direction of order oriented slotting [10].

Since COI is a generally accepted and frequently applied slotting strategy ("put fast movers up front"), it is important to know whether there is any quality guarantee for it in a general multi command situation. This leads us to the question: how bad can the COI performance be? One might think of a finite worst-case ratio such as the one found by Graham [12], who - in a pioneer study - showed that for the problem of scheduling jobs on m identical parallel machines, a list scheduling algorithm always delivers a schedule with makespan no larger than $2\text{-}1/m$ times the optimum. In particular, regardless of the problem dimensions, i.e., the number of machines, list scheduling will always perform no worse than twice the optimum.

So, would there be a worst-case ratio r such that, regardless of the warehouse configuration, COI will always perform no worse than r times the optimum? In this paper we show that for COI there is no such finite worst-case ratio. Worst-case behavior of COI is infinitely bad. We construct a worst-case example that proves the following. Given an arbitrary positive integer p, then there is a warehouse and an order set for that warehouse such that COI when applied to the SKUs belonging to this order set performs p times worse than optimal in total order picking travel time.

The paper is organized as follows. Section 2 is devoted to the construction and analysis of our worst-case example. In Section 3 we give our conclusions.

2 A worst-case example

Crucial in our worst-case example is imbalance in the order set. We construct one huge, popular order and many small, slightly less popular orders. COI grants the SKUs in this huge order a position near the I/O point at the expense of the SKUs in the smaller orders. Before entering into details let us first give a simple example to illustrate our ideas.

Consider the simple rectangular warehouse depicted in Figure 1.

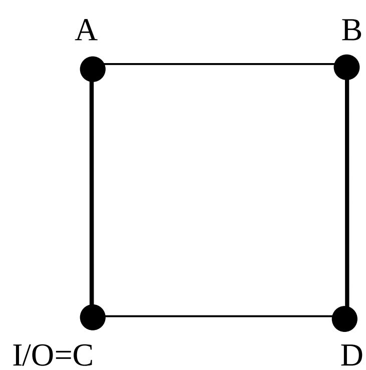

Figure 1: A simple rectangular warehouse with two (vertical) aisles and four pick locations. Dimensions are such that picking e.g. an SKU from location B takes a travel distance of 4.

This warehouse has two parallel aisles at unit distance apart. Each aisle has two pick locations at unit distance. There is one I/O point coinciding with the south-west pick location. Let order picking be done according to the optimal routing policy. Assume that travel time is directly proportional to travel distance. In the warehouse one stores the SKUs 1, 2, 3, and 4. These SKUs have identical dimensions. Each pick location can contain exactly one SKU. On a yearly basis, let the order set for the warehouse be given by:

N times {1}

$N+1$ times {2, 3, 4}

A COI storage assignment places SKU 1 at pick location B. The associated travel distance *L(COI)* to pick all orders is given by:

$$L(COI) = N * 4 + (N+1) * 4$$

An optimal storage assignment places SKU 1 at pick location C. The associated travel distance *L(OPT)* to pick all orders is given by:

$$L(OPT) = (N+1) * 4$$

The limit of *L(COI)/L(OPT)* for *N* tending to infinity is equal to 2.

In this example the large popular order {2, 3, 4} expels the slightly less popular order {1} from any position near the I/O point. This results in a bad travel time performance (nearly twice the optimum).

With the previous example in mind let us now move to our worst-case example which we formulate as follows.

Assertion:

Given an arbitrary positive integer *p*, then there is:

(i) a warehouse configuration
(ii) a set of SKUs
(iii) a set of orders for these SKUs

such that slotting these SKUs in the warehouse according to COI leads to an order picking travel time which is *p* times larger than the order picking travel time produced by the optimal slotting strategy.

Proof:

Let *k* be a positive integer such that $k > p$-1. Let *N* be a positive integer. Consider the simple rectangular warehouse consisting of a single one-dimensional aisle with $k+N$ equidistant pick locations. Let order picking be done according to the optimal routing policy. Let there be one I/O point. Let us parameterize dimensions such that the warehouse can be visualized as exemplified by Figure 2. As usual, we assume that travel time is directly proportional to travel distance.

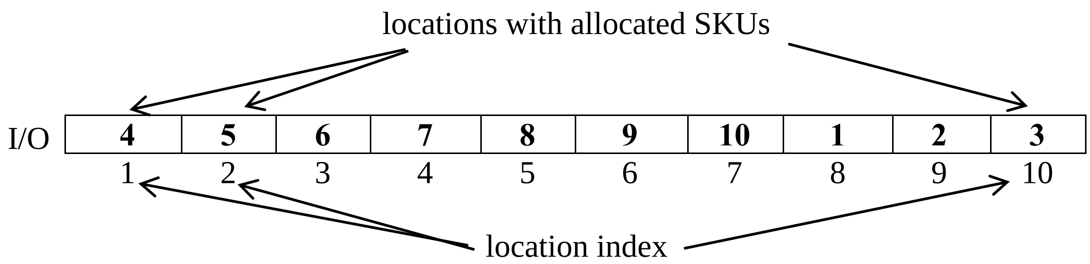

Figure 2: Warehouse layout: a single one-dimensional aisle with equidistant pick locations. Dimensions are such that picking e.g. SKU 7 takes a travel distance of 8.

In the warehouse one stores the SKUs 1,2,…, $k+N$. These SKUs have identical dimensions. Each pick location can contain exactly one SKU. On a yearly basis, let the order set for the warehouse be given by:

N times $\{k\}$
$N+1$ times $\{k\text{-}1\}$
$N+2$ times $\{k\text{-}2\}$.
.
.
.
$N+k$ -2 times $\{2\}$
$N+k$ -1 times $\{1\}$
$N+k$ times $\{k+1, k+2, \ldots, k+N\}$

Apart from a permutation of the leftmost N SKUs, any COI storage assignment is given by Figure 3.

I/O	**k+1**	**k+2**	**…**	**k+N**	**1**	**2**	**…**	**k**
	1	2	…	N	N+1	N+2	...	N+k

Figure 3: COI storage assignment

The associated halfway travel distance $L(COI)$ is given by:

$$L(COI) = N(N+k) + \sum_{j=1}^{k} (N+k-j)(N+j) = (k+1)(6N^2 + 6Nk + k^2 - k)/6$$

Now, consider the storage assignment *OPT* given by Figure 4, which is easily seen to be optimal using an exchange argument.

I/O	**1**	**2**	**…**	***k***	***k*+1**	***k*+2**	**…**	***k*+*N***
	1	2	…	*k*	*k*+1	*k*+2	…	*k*+*N*

Figure 4: Optimal storage assignment

The associated halfway travel distance *L(OPT)* is given by

$$L(OPT) = (N+k)^2 + \sum_{j=1}^{k} j(N+k-j) = (N+k)^2 + k(k+1)(3N+k-1)/6$$

For fixed *k*, the limit of *L(COI)/L(OPT)* for *N* tending to infinity is equal to $k+1$. Since $p < k+1$ it holds that $L(COI) > p\, L(OPT)$ for *N* large enough. ■

In the above example the orders - apart from their multiplicity - are disjoint. Hence, one might argue that there is a simple trick to let COI work properly. Redefine the set $\{k+1, k+2, \ldots, k+N\}$ by treating it as one SKU occupying *N* pick locations. Hence, its COI becomes $N/N+k$ which - for *N* large enough - is larger than the other COIs. So we find an optimal storage assignment as in Figure 4. Our example can easily be adapted so as to avoid this trick. Just add one order {k+1} with multiplicity 1. Then the orders are not disjoint, so the trick will not work. COI and optimal storage assignments are the same as in Figure 3 and 4, respectively. Also, the limit behavior of *L(COI)/L(OPT)* does not change.

3 Conclusions

This paper shows that for the Cube per Order Index (COI) slotting strategy there is no finite worst-case ratio with respect to order picking travel time. We construct a worst-case example that proves the following. Given an arbitrary positive integer *p*, then there is a warehouse and an order set for that warehouse such that COI when applied to the SKUs belonging to this order set performs *p* times worse than optimal in total order picking travel time.

The contribution of our paper lies in finding a worst-case example. In our view, this is nontrivial. Of course, we also prove that our example is indeed worst-case. However, the proof is simple.

Crucial in our worst-case example is imbalance in the order set. We construct one huge, popular order and many small, slightly less popular orders. COI grants the SKUs in this huge order a position near the I/O point at the expense of the SKUs in the smaller orders.

In practice, imbalanced order sets may cause the COI slotting strategy to perform poorly as well. Of course, the poorness will be limited for the following reason. In practice, one works with a fixed warehouse. So, for that particular warehouse, there is a finite worst-case ratio for COI: order picking travel time will be no larger than a fixed number - depending on the layout dimensions of the warehouse - times the optimum. Evidently, this fixed number is bounded from above by the - routing-specific - order picking travel time required to pick all SKUs in the warehouse in one tour.

Obviously, when deciding upon the right slotting strategy, carefully scrutinizing the order structure is imperative. When single command retrievals are dominant, then COI is an excellent candidate. On the other hand, this paper shows how bad things can be when a single command slotting strategy is implemented for a multi command situation.

Acknowledgement

I would like to convey my thanks to warehousing guru Ronald Mantel for numerous inspiring discussions about the idiosyncrasies of the COI slotting strategy.

References

[1] Heskett, J. L., “Cube-per-Order Index - A Key to Warehouse Stock Location,” *Transport and Distribution Management*, 3, 27-31 (1963).

[2] Heskett, J. L., “Putting the Cube-per-Order Index to Work in Warehouse Layout,” *Transport and Distribution Management*, 4, 23-30 (1964).

[3] Heskett, J. L., *Business Logistics: Management of Physical Supply and Distribution*, Ronald Press Co. (1964).

[4] Kallina, C. and Lynn, J., “Application of the Cube-per-Order Index Rule for Stock Location in a Distribution Warehouse,” *Interfaces*, 7 (1), 37-46 (1976).

[5] Malmborg, C. J. and Bhaskaran, K., “On the Optimality of the Cube per Order Index for Conventional Warehouses with Dual Command Cycles,” *Material Flow*, 4, 169-175 (1987).

[6] Malmborg, C. J. and Bhaskaran, K., “Optimal Storage Assignment Policies for Multiaddress Warehousing Systems,” *IEEE Transactions on Systems, Man and Cybernetics*, 19 (1), 197-204 (1989).

[7] Malmborg, C. J. and Bhaskaran, K., “A Revised Proof of Optimality for the Cube-per-Order Index Rule for Stored Item Location,” *Applied Mathematical Modelling*, 14 (2), 87-95 (1990).

[8] Malmborg, C. J., "Optimization of Cubic-per-Order Index Layouts with Zoning Constraints," *International Journal of Production Research*, 33 (2), 465-482 (1995).

[9] Malmborg, C. J., "Storage Assignment Policy Tradeoffs," *International Journal of Production Research*, 34 (2), 363-378 (1996).

[10] Mantel, R. J., Schuur, P. C. and Heragu, S. S., "Order Oriented Slotting: A New Assignment Strategy for Warehouses," *European Journal of Industrial Engineering*, 1 (3), 301-316 (2007).

[11] De Ruijter, H., Schuur, P. C., Mantel, R. J. and Heragu, S. S., "Order Oriented Slotting and the Effect of Order Batching for the Practical Case of a Book Distribution Center," *Proceedings of the 2009 ICOVACS Conference*, October 19-21, Louisville, KY (2009).

[12] Graham, R. L., "Bounds for Certain Multiprocessing Anomalies," *Bell System Technological Journal*, 45, 1563-1581 (1966).