Analysis of Batching for a Carousel in a Production Environment

B. Rouwenhorst[a], G.J. van Houtum[b,*], W.H.M. Zijm[b]

[a] University of Twente, Faculty of Mechanical Engineering

[b] Eindhoven University of Technology, Faculty of Technology Management

March 1999

Abstract

When using a carousel in a production environment, it is required that the orders for products receive sufficiently low response times from the carousel. The response times strongly depend on the required service time per order. The service times consist of pure picking time and an amount of time needed to rotate the carousel to the right positions. For the latter part, the average amount of time per order can be strongly reduced by picking the orders in batches instead of on an individual basis. So, on one hand, batching has a reducing effect on the response times. On the other hand, it has an increasing effect on the response times, since it introduces additional times to form the batches. By performing a stochastic analysis, we derive explicit formulae for the mean response time as a function of the batch size. These formulae can be easily exploited to determine the optimal batch size, and to generate figures that show the behavior of the optimal batch size as a function of the arrival rate of orders and the behavior of the mean response time as a function of this arrival rate when for each rate the optimal batch size is used.

Keywords: Carousel, batching, queueing analysis

* *Corresponding author.* Mailing address: Eindhoven University of Technology, Faculty of Technology Management, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. Phone: +31 40 2475163. Fax: +31 40 2464596. E-mail: g.j.v.houtum@tm.tue.nl .

1 Introduction

A carousel is a computer controlled warehousing system used for storage and order picking of small to medium sized products. Such a system may have many different products stored in bins that rotate in a closed loop. The system is operated by an order picker (either human or robotic) which occupies a fixed position in front of the carousel. Because the system automatically rotates the bin containing the requested product to the position of the order picker, it is referred to as a *product-to-picker* system. The order picker may effectively use the rotation time of the carousel for activities such as sorting, packaging or labeling of the retrieved goods.

Two types of carousel systems are used in practice: the horizontal and the vertical carousel. The horizontal carousel consists of a large number of carriers driven by a powered chain that can rotate around a narrow closed loop. Each carrier consists of multiple bin positions; see Figure 1. The vertical carousel, also called paternoster, consists of a number of shelves which at the ends are connected with vertically rotating chains. Each shelf carries multiple bins. Most paternosters are contained in a closed metal cage with only one opening at the position of the order picker. This allows for protection of the stored products. Most human-operated horizontal carousels are not higher than two meters. The length of such a carousel can be much longer, up to 50 meters, say.

The horizontal carousel is usually found in distribution environments where its main advantage is the reduction of the work load of the order pickers by reducing travel time. The paternoster is often used in production environments, in order to save floor space. In this case the paternoster can be on, or close to, the production floor instead of in a distant warehouse.

In all types of warehouses, efficiency is important: the orders must be retrieved at minimal costs. This implies that a warehouse system should have a large maximum throughput per day (*capacity*). In this paper we focus on carousels used for storing products that are needed for production or assembling. This imposes an extra restriction for the warehouse system: the items must be retrieved fast (i.e. without long waiting times). It usually also implies that orders consist of a small variety of articles (*line items*), and it implies that orders arrive at the carousel system (seemingly) randomly during the working period.

A large part of order picking time consists of traveling or rotation time. A way to decrease travel time per order is to combine several orders. In this way, one big order is created, that can be retreived in a more efficient way than when all orders are picked one by one. This is called *batching* and the number of orders per batch is called the *batch size*. Batching has a decreasing effect on the retrieval time per order, and thus an increasing effect on the capacity. Apart from retrieving, the products have to be sorted per order. It is assumed that this sorting process is done during the rotation time of the carousel and does not require extra time. This way of sorting is called *sort-while-pick*.

The response time of an order is defined as the total time between the order arrival and the order completion. This time consists of three elements. The first is

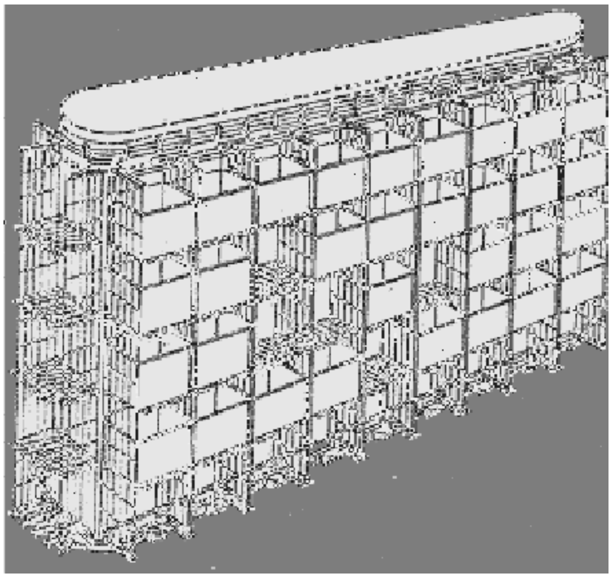

Figure 1: A horizontal carousel.

the *order waiting time*, during which the orders wait until there are enough orders to make a batch of a certain size. If the batch size is one, this time will be zero. After this order waiting time, there can be a *batch waiting time*, when the batch has to wait until the resource (the carousel) is available. Finally, the batch will be processed during the *batch service time* of the carousel. After this the batch, or better, each of the orders in the batch, leaves the system.

A large number of orders per batch gives a large capacity of the carousel. On the other hand, a large number of orders per batch lets the orders wait long until a batch can be formed. In order to minimize the response time for an arbitrary order, the optimal balance between these two effects should be determined.

The research object of this paper is a carousel system used for storing products that are needed for production or assembling. The input is formed by the system and order characteristics. Further, the batch size is a control parameter. The aim of this paper is to develop a method to forecast the behavior of the system, for given values of both the input and control parameters. With this method, it must be possible to find the batch size that minimizes the mean response time for the given input. The main result is an accurate explicit approximation of the resulting response time.

1.1 Related Literature

Related literature can be divided into two parts: literature on the performance of a carousel and literature on batching in other storage systems. We are not aware of any literature on the combination of batching and carousel systems.

Carousel systems in general have received considerable attention in the literature. Bartoldi and Platzman [1] and Stern [7] consider the single order pick sequencing problem. This problem concerns the sequence of the picks, within an order, that has to be optimized in order to get a short route. They present a linear time algorithm that finds an optimal solution. Sharp et al. [6] derive a formula for the minimum order pick time of a carousel system, but they consider only two sequences. Rouwenhorst et al. [5] present an approximation for the minimum order pick time when all candidate sequences are considered.

Van den Berg [8] and Ghosh and Wells [3] present dynamic programming algorithms that find an optimum sequence of the picks when orders have to be retrieved one by one, in a fixed sequence. Bartoldi and Platzman [1] and Van den Berg [8] also consider the same problem under the assumption that orders may be processed in any arbitrary sequence.

The question if and how to batch is examined for another system, viz. an aisles-based order pick system, by Choe [2]. His approach is similar to our approach.

1.2 Organization of the Paper

Section 2 describes the model. The model consists of three parts: the order queue, the batch queue and the carousel server. For each part, explicit (approximate) expressions are presented for the mean response time of an arbitrary order. In particular, these expressions use the first two moments of the rotation time of the carousel. Section 3 focuses on this time. Numerical results will be included in a next version of this paper.

2 Model

This section describes the model of the carousel and its environment.

2.1 The Environment of the Carousel

The environment of the carousel can be characterized by describing the arrival stream and the number of line items per order. It is assumed that orders arrive according to a renewal process with generally distributed interarrival times. The random variable T_a is used to denote an arbitrary interarrival time. The mean and the coefficient of variation of the interarrival times are denoted by $\mu_a = \mathbb{E}T_a$ and $c_a = \sqrt{\mathbb{E}(T_a - \mathbb{E}T_a)^2}/\mathbb{E}T_a$. These two variables can be derived from the expected number of orders per day μ_d, and its coefficient of variation c_d by:

$$\mu_a = \frac{1}{\mu_d} \text{ and } c_a = \sqrt{\frac{c_d^2}{\mu_d^3}}\ . \qquad (1)$$

It is assumed that each order consists of a random number of line items that is denoted by the generic random variable N. Each item stored in the carousel has an equal chance to be requested. The probability distribution of the number of line items per order is given by:

$$p_n = Pr(N = n) \ , \ n \geq 1. \tag{2}$$

Orders are handled in the form of batches. The batch size b is a control parameter. The probability that the number of order lines in a batch of size b is equal to n, is denoted by $p_n^b, n \geq 1$. The probability that order lines of different orders coincide is neglected, since most carousel systems consist of a large number of carriers. By definition, $p_n^1 = p_n$ for all $n \geq 1$. For batch sizes larger than 1, the probabilities p_n^b are calculated by:

$$p_n^{b+1} = \sum_{i=1}^{n-1} p_{n-i}^b p_i \ , \qquad n \geq 1, \ b \geq 1. \tag{3}$$

The characteristics of the carousel are the (fixed) time needed for one complete rotation of the carousel, denoted by t_{rot}, and the first two moments of the extraction time T_{ext}. The extraction time presents the time during which a line item is extracted from the carousel as well as the extra time needed for acceleration and deceleration of the carousel at each stop.

The model consists of three parts, as depicted in Figure 2: The order queue, the batch queue and the server.

Orders subsequently visit each part and spend W_o, W_b and S_b time units in the order queue, the batch queue and the carousel, respectively. The total duration gives the response time T:

$$T = W_o + W_b + S_b \ . \tag{4}$$

The three components are further discussed in the remainder of this section.

2.2 The Order Queue

An order first arrives at the order queue where it waits to be enclosed in a batch, consisting of b orders. The mean waiting time in the order queue is given by:

$$\mathbb{E}W_o = \frac{b-1}{2\mu_d} = \frac{b-1}{2}\mu_a \ . \tag{5}$$

This waiting time will dominate if the orders do not arrive very frequently and b is high.

2.3 The Batch Queue

After visiting the order queue, the order, being part of a batch, arrives in the batch queue. The batches arrive according to a renewal process with a mean and coefficient of variation of the interarrival times equal to $\mu_b = b\mu_a$ and $c_b = c_a/\sqrt{b}$, respectively. The retrieval of this batch starts as soon as the carousel becomes available. Let S_b denote the random retrieval (service) time of a batch. Then the utilization rate of

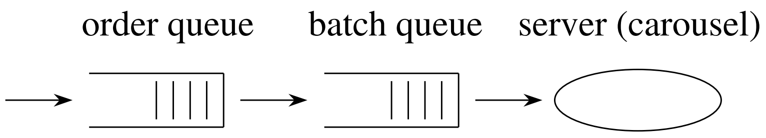

Figure 2: The model.

the carousel is equal to $\rho = \mu_b \mathbb{E} S_b$. The coefficient of variation of S_b is denoted by c_b.

The batch queue combined with the server can be seen as a GI/G/1 queue. The corresponding batch waiting time W_b dominates if the carousel is heavily utilized. This may happen if the batch size is rather small. The system might not even be ergodic if the batch size is too small. The waiting time in case of ergodicity is approximated by a formula developed by Kreamer and Langenbach-Belz [4]. In our case, we obtain

$$W_b = g(\rho, c_b^2, c_s^2) \left(\frac{c_b^2 + c_s^2}{2} \right) \frac{\rho}{1-\rho} \mathbb{E} S_b \ , \tag{6}$$

where $g(\rho, c_a^2, c_s^2)$ represents a correction factor that is equal to

$$g(\rho, c_b^2, c_s^2) = \begin{cases} \exp\left(\frac{-2(1-\rho)}{3\rho} \frac{(1-c_b^2)^2}{(c_b^2+c_s^2)} \right) & , \quad c_b^2 \leq 1, \\ \exp\left(-(1-\rho) \frac{(c_b^2-1)}{(c_b^2+4c_s^2)} \right) & , \quad c_b^2 > 1. \end{cases}$$

This approximation is especially suited for the case that c_b^2 is much smaller than 1. If the orders arrive at the order queue according to a Poisson process, the interarrival times of the batches will have an Erlang(b) distribution with a squared coefficient of variation equal to $1/b$. Therefore it is likely that the approximation will be suited for our model.

Formula (6) gives the expected waiting time as a function of, among others, the first and second moment of the retrieval time of a batch. These two moments depend on the server characteristics and will be further analyzed in the next subsection.

2.4 The Server

After visiting the order queue and the batch queue, the orders, included in a batch, will be retrieved from the carousel. The retrieval time of a batch consists of rotation times and extraction times, during which the line items are physically removed from the carousel. It is assumed that the rotation time between line items does not depend upon the height of the product locations. The *normalized rotation time* D_n is defined as the fraction of one complete rotation that the carousel has to rotate during the retrieval of n line items. It is also assumed that the extra time that is needed to stop and accelerate the carousel before and after an extraction is included in the *order pick time per pick* T_{ext}. This, combined with the *time needed for one complete rotation of the carousel*, denoted by t_{rot}, yields the *retrieval (service) time of a batch consisting of* n *line items* according to

$$\hat{S}_n = n \cdot T_{ext} + t_{rot} D_n \ . \tag{7}$$

Then $S_b = \hat{S}_n$ with probability p_n^b. Using (7), we find

$$\mathbb{E}S_b = \sum_{n\in\mathbb{N}} p_n^b(n\mathbb{E}T_{ext} + t_{rot}\mathbb{E}D_n) = \mathbb{E}N_b\mathbb{E}T_{ext} + t_{rot}\sum_{n\in\mathbb{N}} p_n^b\mathbb{E}D_n \ , \tag{8}$$

$$\begin{aligned}\mathbb{E}S_b^2 &= \sum_{n\in\mathbb{N}} p_n^b(n\mathbb{E}T_{ext} + t_{rot}\mathbb{E}D_n)^2 \\ &= \mathbb{E}N_b^2(\mathbb{E}T_{ext})^2 + 2t_{rot}\mathbb{E}T_{ext}\sum_{n\in\mathbb{N}} p_n^b n\mathbb{E}D_n + t_{rot}^2\sum_{n\in\mathbb{N}} p_n^b(\mathbb{E}D_n)^2 \ .\end{aligned} \tag{9}$$

where N_b denotes the number of line orders of a complete batch. So, $\mathbb{E}N_b = b\mathbb{E}N$ and $\mathbb{E}N_b^2 = var(N_b) - (\mathbb{E}N_b)^2 = b\ var(N) - b^2(\mathbb{E}N)^2$.

Note that, in the formulas (8) and (9), the numerical evaluation of the sums over $n \in \mathbb{N}$ is straightforward if the index set is finite. If the index is infinite, then the probabilities p_n and p_n^b will become very small for large values of n and an efficient numerical evaluation of these sums is still possible. The random variable D_n remains to be analyzed.

Concluding, after dividing the model in three parts, expressions were derived for the waiting times in the first two parts. For the third part, the server, the number of unknowns was reduced to one, which is the normalized rotation time D_n. Its first and second moment are needed for the analysis of the response time. The derivation of these moments comes from an earlier study; see Rouwenhorst et al. [5].

3 The Normalized Rotation Times D_n

This section summarizes the analysis of the normalized rotation times D_n.

3.1 The Carousel Model

The carousel can be depicted as a circle with circumference 1. The locations of the line orders are supposed to be uniformly distributed on the circle, while a given fixed point denotes the location of the picker; see Figure 3.

3.2 The Retrieval Sequence

To retrieve a batch, the carousel has to rotate the different locations in front of the order picker. The sequence in which the products are retrieved is assumed to be not fixed. Therefore, the best sequence can be chosen, which will be the shortest possible sequence that visits all locations. Bartoldi and Platzman [1] and Stern [7] have analyzed the optimal retrieval sequence, and found that the best sequence is among $2n$ sequences, when n locations have to be visited. To make such a sequence more visible, it is pretended that the order picker travels around the carousel instead of the other way round. Four different sequences are visualized in Figure 4. Sequence

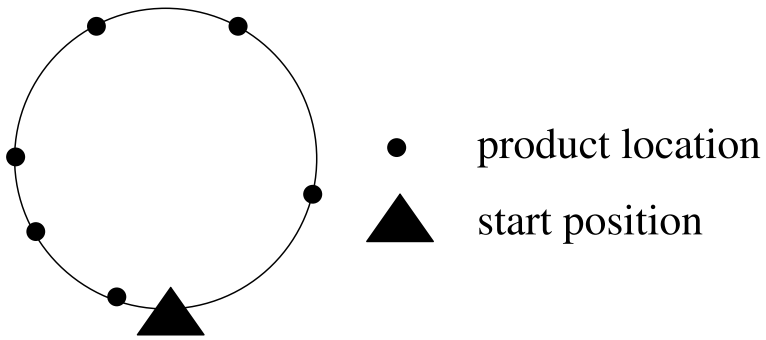

Figure 3: Graphical representation of the carousel.

1 lets the picker visit the locations in the clockwise direction. Sequence 3 lets the order picker travel to the nearest location in the counterclockwise direction, and then visits the remaining locations in the clockwise direction. In general, sequence $2i-1$, $i = 1, \ldots, n$, lets the order picker travel to the i-th nearest location in the counterclockwise direction, and then visits the remaining locations in the clockwise direction. The sequences $2, 4, \cdots$ are mirrored from the sequences $1, 3, \cdots$.

Further analysis (see [5]) shows that it is likely that the shortest route is formed by a sequence with a low number. This observation will be used in the next subsection.

3.3 Accurate Approximation for the D_n

When n locations are to be to visited, $2n$ routes are to be considered, to find the shortest one. But we can not determine the distribution function of the rotation time of the overall shortest route. However, we can determine the distribution function of the shortest of a limited number of sequences (see [5]). This gives the following results.

For $n = 1$ and $n = 2$ the results are equal to the overall shortest sequence. For $n = 3$ and $n = 4$, the shortest of the first four four sequences has been derived. For $n \geq 5$ the shortest of the first six sequences has been determined. This leads to upper bounds for the first two moments of D_n as depicted in Table 1.

The tightness of the upper bounds has been tested by simulation. It has been shown that the upper bounds are very tight for all values of n. Hence, we use them as approximations for the first two moments of the D_n.

This completes the analysis of the carousel for a given batch size. Next steps consist of the determination of the optimal batch size and the generation of figures of the types mentioned at the end of the abstract of this paper.

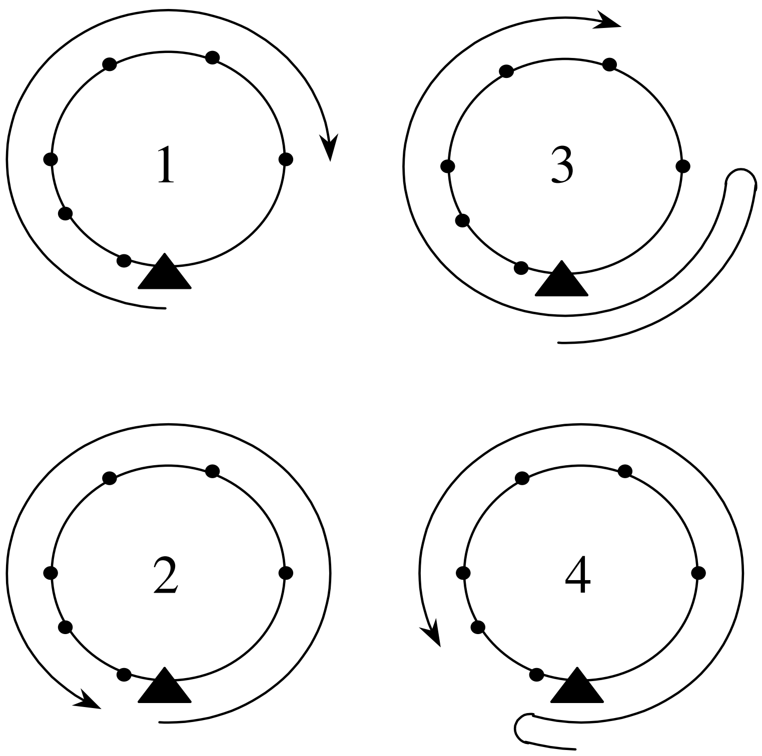

Figure 4: The routes defined by the sequences 1-6.

n	ED_n	ED_n^2
1	$\frac{1}{4}$	$\frac{1}{12}$
2	$\frac{5}{12}$	$\frac{19}{96}$
3	$\frac{17}{32}$	$\frac{175}{576}$
4	$\frac{5}{8}$	$\frac{355}{864}$
≥ 5	$\frac{4480n-4593}{4480(n+1)}$	$\frac{806400n^2-847080n-548357}{806400(n+2)(n+1)}$

Table 1: Upper bounds for the D_n.

References

[1] Bartholdi III, J.J., and Platzman, L.K. Retrieval strategies for a carousel conveyor. *IIE Transactions 18* (1986), pp. 166–173.

[2] Choe, K.I. *Aisle-based Order Pick Systems with Batching, Zoning and Sorting.* Ph.D. Thesis, Georgia Institute of Technology, Atlanta, 1991.

[3] Ghosh, J.B., and Wells, C.E. Optimal retrieval strategies for carousel conveyors. *Mathematical Computer Modelling 16* (1992), pp. 59–70.

[4] Kraemer, W., and Langenbach-Belz., M. Approximate formulae for the delay in the queueing system $GI|G|1$. In *Proceedings of the 8th International Teletraffic Congress*, Melbourne, 1976, pp. 235– .

[5] Rouwenhorst, B., Van den Berg, J.P., Van Houtum, G.J., and Zijm, W.H.M. Performance analysis of a carousel system. In *Progress in Material Handling Research: 1996*, The Material Handling Industry of America, Charlotte, NC, 1996, pp. 495–511.

[6] Sharp, G.P., Eckert, J., and Gibson, D. Order picking using horizontal carousels. Research report, Georgia Institute of Technology, Atlanta, 1990.

[7] Stern, H.I. Parts location and optimal picking rules for a carousel conveyer automatic storage and retrieval system. In *7th International Conference on Automation in Warehousing*, 1986, pp. 185–193.

[8] Van den Berg, J.P. Multiple order pick sequencing in a carousel system: A solvable case of the Rural Postman Problem. *Journal of the Operational Research Society 47* (1996), pp. 1504–1515.