ANALYZING ORDER THROUGHPUT TIMES IN A MILKRUN PICKING SYSTEM

J.P. van der Gaast
Erasmus Universiteit Rotterdam

M.B.M. de Koster
Erasmus Universiteit Rotterdam

I.J.B.F. Adan
Technische Universiteit Eindhoven

Abstract

E-commerce fulfillment competition evolves around cheap, speedy, and time-definite delivery. Milkrun order picking systems have proven to be very successful in providing handling speed for a large, but highly variable, number of orders. In this system, an order picker picks orders that arrive in real time during the picking process; by dynamically changing the stops on the picker's current picking route. The advantage of milkrun picking is that it reduces order picking set-up time and worker travel time compared to conventional batch picking systems. This paper is the first in studying order throughput times of multi-line orders in a milkrun picking system. We model this system as a cyclic polling system with simultaneous batch arrivals and determine the mean order throughput time. These results allow us to study the effect of different product allocations. For a real world application we show that milkrun order picking reduces the order throughput time significantly compared to conventional batch picking.

1 Introduction

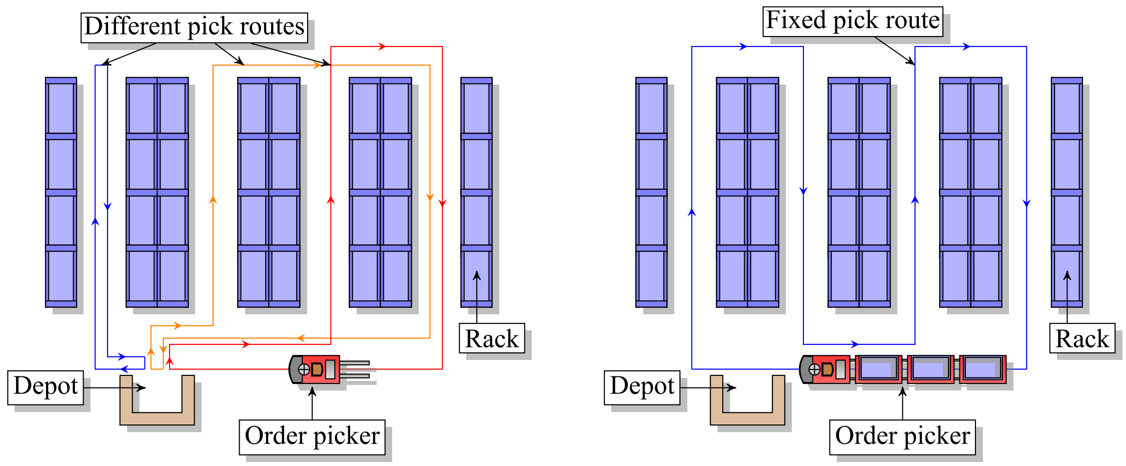
Recent technological advances and trends in distribution and manufacturing have led to a growth in complexity of warehousing systems. Today's warehouse operations face challenges like the need for shorter lead times, for real-time response, to handle a larger number of orders with greater variety, and to deal with flexible processes [4].

Batch picking is a common way to organize the picking process, where daily a large number of customer orders needs to be picked (Figure 1a). Batch picking is a picker-to-parts order picking method in which the demand from multiple orders is used to form so-called pick batches [1]. Pick routes are constructed for each pick batch to minimize the total travel time of the order picker (see e.g. Gademann & Van de Velde [2]). A drawback of this approach is that batch formation takes time, and, as customers demand shorter lead times, more efficient ways to organize the order picking process exist. In this paper we study an alternative method of order picking, which we denote by *milkrun picking*, that allows shorter order throughput times compared to conventional batch picking systems, in particular for high order arrival rates.

In a milkrun picking system (Figure 1b), an order picker picks orders in batches that arrive in real-time and integrates them in the current picking cycle. This subsequently changes dynamically the stops on the order picker's picking route [3]. The picker is constantly traveling a fixed route along the aisles of a part or the entire order picking area. Using modern order-picking aids like pick-by-voice techniques or by a handheld terminal, new pick instructions are received continuously and are included in the current picking cycle. In case the lines of an incoming customer order are located either at the current stop or further downstream in the picking route, the picker can pick this order in the current picking cycle. In a traditional batch picking system, an incoming customer order would only be picked in one of the following picking cycles. After the picking cycle has been completed and the order picker reaches the depot, the picked products are disposed and sorted per customer order (i.e. using a pick-and-sort system), and a new picking cycle starts immediately. This way of order picking saves set-up time, worker travel time, and allows fast customer response, particularly for high order arrival rates which are often experienced in warehouses of e-commerce companies [4]. In addition, short order throughput times are important as e-commerce companies are inclined to set their order cut-off times as late as possible while still guaranteeing that orders can be delivered next day or in some cases even the same day.

In this paper, we study the mean order throughput time in a milkrun picking system, i.e. the time between a customer order entering the system until the whole order is delivered at the depot. The order throughput time strongly depends on the product (or storage) allocation in the order picking area. Typically, an incoming customer order consists of one or more order lines, each for a product stored at a different location within the order picking area. Therefore, in order to achieve short order throughput times products should be allocated in an optimal way in order to increase the probability that an incoming customer order can be included and fully picked in the current picking cycle. We study for a real world application the mean order throughput time, as well as the effect of different product allocations. Our results will help both designers and managers to create optimal design and control methods to improve the performance of a milkrun picking system.

The organization of this paper is as follows. In section §2 a detailed description of the



(a) Batch picking

(b) Milkrun picking

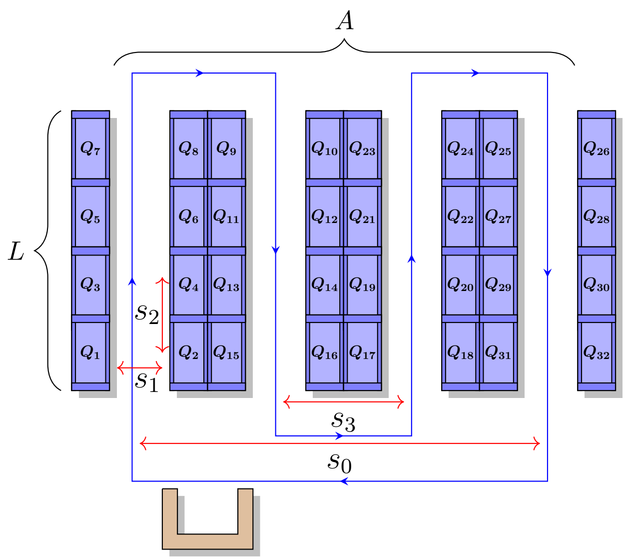
Figure 1: Comparison of batch and milkrun picking.

model and the corresponding notation used in this paper are given. Next, section §3 provides the analysis of the mean order throughput times. We extensively analyze the results of our model in section §4 for a real world application. Finally, in section §5 we conclude and suggest some extensions of the model and further research topics.

2 Model description

Consider a milkrun picking system as shown in Figure 2. We assume the order picking area to have a parallel aisle layout, with A aisles and L storage positions on each side of an aisle (a rack). Within an aisle, a single order picker applies two-sided picking, i.e. simultaneous picking from the right and left sides within an aisle. We denote the storage locations by $Q_1, \ldots, Q_N$, where the number of storage locations N equals $2AL$. Each storage location can be considered as a queue for order lines requesting the product stored on that location. Without loss of generality, we assume that the number of storage locations equals the number of different products stored in the warehouse. For the ease of presentation, all references to queue indices greater than N or less than 1 are implicitly assumed to be modulo N, e.g., Q_{N+1} is understood as Q_1. The order picker visits all queues according to a strict S-shape routing policy in a cyclic sequence and picks all required products for the outstanding customer orders to a pick cart or tow-train. This means that every aisle is completely traversed during a picking cycle, because new customer orders can enter the system in real-time. Therefore, the order picker cannot skip entering an aisle like in conventional batch picking. We assume the number of products the order picker can pick per picking cycle is unconstrained, as for online retailers the route often finishes before the cart or train is full [3]. This implies that every customer order is either fully picked by the end of the current cycle or at the end of the next cycle. Finite capacity of the pick cart and storing the same product at multiple locations are considered to be further extensions of the model.

A milkrun picking system with multi-line customer orders arriving in real-time can be accurately modeled using a polling system with simultaneous batch arrivals [6]. Polling systems are multi-queue systems served by a single server who cyclically visits the queues in order to serve the customers waiting at these queues. Typically, when moving from one queue to another the server incurs a switch-over time. In a milkrun picking system, the order picker is represented by the server and a storage location by a queue, and a multi-line order represents multiple simultaneously arriving customers (a batch).



A := number of aisles
L := number of storage positions in a rack
s_0 := return travel time and depot time
s_1 := travel time side to side
s_2 := travel time within aisles between adjacent storage location
s_3 := travel time between adjacent aisles

Figure 2: Overview of the milkrun picking system.

New customer orders arrive at the system according to a Poisson process with rate λ. Each customer order is of size $\boldsymbol{D} = (D_1, \dots, D_N)$, where D_j, $j = 1, \dots, N$ represents the number of units of product j is requested. Let $\boldsymbol{K} = \Phi(\boldsymbol{D})$, where $\Phi : \mathbb{N}^N \to \mathbb{N}^N$. Mapping Φ defines the product allocation of the products to their storage locations and is given by, $\Phi(\boldsymbol{D}) = \boldsymbol{D}x$, where $x_{ij} \in \mathbb{N}^{N \times N}$ with $x_{ij} = 1$ if product j is allocated to storage location i and 0 otherwise. Then, for each order, $\boldsymbol{K} = (K_1, \dots, K_N)$, where K_i represents the number of units that need to be picked at Q_i, $i = 1, \dots, N$ for that order. The random vector $\boldsymbol{K}$ is assumed to be independent of past and future arriving epochs and for every realization at least one product needs to be picked. The support with all possible realizations of $\boldsymbol{K}$ is denoted by $\mathcal{K}$, and we denote by $\boldsymbol{k} = (k_1, \dots, k_N)$ a realization of $\boldsymbol{K}$. The joint probability distribution of $\boldsymbol{K}$ is denoted by $\pi(\boldsymbol{k}) = \mathbb{P}(K_1 = k_1, \dots, K_N = k_N)$. The arrival rate of product units that need to be picked at Q_i is denoted by $\lambda_i = \lambda E(K_i)$. The total arrival rate of products to be picked for the customer orders arriving in the system is given by $\Lambda = \sum_{i=1}^{N} \lambda_i$. The order throughput time of an arbitrary customer order is denoted by T and is defined as the time between its arrival epoch until the order has been fully picked and delivered at the depot.

At each queue, the picker picks the product units on a First-Come First-Served (FCFS) basis. We assume the order picker picks all product units at the current queue until no product units need to be picked anymore. This also includes demand for the product that arrives while the picker is busy picking at this queue (*exhaustive strategy*). The picking times of a product unit in Q_i dentored by generally distributed random variable B_i are assumed to be independent and identically distributed random variables with first and second moment $E(B_i)$ and $E(B_i^2)$, respectively. The workload at Q_i, $i = 1, \ldots, N$ is defined by $\rho_i = \lambda_i E(B_i)$; the overall system load by $\rho = \sum_{i=1}^{N} \rho_i$. For the system to be stable a necessary and sufficient condition is that $\rho < 1$ [5], which is assumed to be the case in the remainder of this paper.

When the order picker moves from Q_i to Q_{i+1}, he or she takes a generally distributed travel time S_i with first and second moment $E(S_i)$ and $E(S_i^2)$. Without loss of generality, we assume that the travel times from side to side within an aisle are independent and identically distributed with mean s_1 and second moment s_1^2, the travel times within aisles between two adjacent storage locations have mean s_2 and second moment s_2^2, whereas the time required to travel from one aisle to the next one has mean s_3 and second moment s_3^2. Finally, after visiting the last queue the order picker returns to the first queue to start a new cycle. On the way, the order picker visits the depot where he or she will drop off the picked products so that other operators can sort and transport them. We assume that this time is independent of the number of products picked, and it is included in s_0 and its second moment s_0^2. See also Figure 2.

We define a picking cycle from the service beginning at the first queue until the order picker has delivered all the picked products at the depot and arrives at the first queue again. Therefore, a picking cycle C consists of N visit periods, V_i, each followed by a travel time S_i. A visit period V_i starts with a pick of a product unit and ends after the last product has been picked given that product units need to be picked at Q_i. Then, the order picker travels to the next picking location of which the duration is S_i. In case no product units need to be picked at Q_i the order picker immediately travels to next picking location. The total mean duration of a picking cycle is independent of the queues involved and is given by (see, e.g., Takagi [5]) $E(C) = E(S) / (1 - \rho)$. Finally, we assume replenishment is not required in a picking cycle, and each queue has infinite capacity (i.e. no limit on the maximum number of order lines waiting to be picked).

Whether a customer order is fully picked in the picking cycle during which it arrives, or otherwise in the next cycle depends on the location of the server and the picking strategy. Therefore, let $\mathcal{K}_j^0$ and $\mathcal{K}_j^1$, $j = 1, \ldots, N$ be subsets of support $\mathcal{K}$, defined as

$$\mathcal{K}_j^0 = \{k_1 = 0, \ldots, k_{j-1} = 0, k_j \geq 0, k_{j+1} \geq 0, \ldots, k_N \geq 0\} \in \mathcal{K},$$

and $\mathcal{K}_j^1 = \left(\mathcal{K}_j^0\right)^c$ as its complement such that for $j = 1, \ldots, N$ we have $\mathcal{K}_j^0 \cup \mathcal{K}_j^1 = \mathcal{K}$

and let the associated probabilities be $\pi\left(\mathcal{K}_j^0\right)$ and $\pi\left(\mathcal{K}_j^1\right)$. The interpretation of $\boldsymbol{k} \in \mathcal{K}_j^0$ is that for an incoming customer order all the products need to be picked at $Q_j, \ldots, Q_N$. For example, this means if the order picker is at Q_j or has not reached Q_j yet a customer order $\boldsymbol{k} \in \mathcal{K}_j^0$ can be included in the current picking cycle, whereas if $\boldsymbol{k} \in \mathcal{K}_j^1$ the order will be completed in the next cycle. Finally, let $E\left(K_i|\mathcal{K}_j^0\right)$ and $E\left(K_i|\mathcal{K}_j^1\right)$ be the conditional mean number of product units that need to picked in Q_i, $i = 1, \ldots, N$ given subset $\mathcal{K}_j^0$ or $\mathcal{K}_j^1$.

3 Mean order throughput time

In order to derive the mean order throughput time, we apply the *Mean Value Analysis* (MVA) of Van der Gaast et al. [6] for the exhaustive strategy. In this MVA a set of N^2 linear equations is derived for calculating $E\left(\bar{L}_i^{(S_{j-1},V_j)}\right)$, the conditional mean queue-length at Q_i (excluding the potential product unit that is being picked) at an arbitrary epoch within travel period S_{j-1} and visit period V_j. These MVA equations for the exhaustive strategy are given in Van der Gaast et al. [6]. With use of the conditional mean queue-lengths, the performance statistics such as the waiting time of a customer can be determined, but also the mean order throughput time as we will show in this section.

For notation purposes we introduce θ_j in this section as shorthand for intervisit period (S_{j-1}, V_j); the mean duration of this period $E(\theta_j)$ is given by, $E(\theta_j) = E(S_{j-1}) + E(V_j)$, $j = 1, \ldots, N$, where $E(V_j) = \rho_j E(C)$ and $\sum_{j=1}^{N} E(\theta_j) = E(C)$.

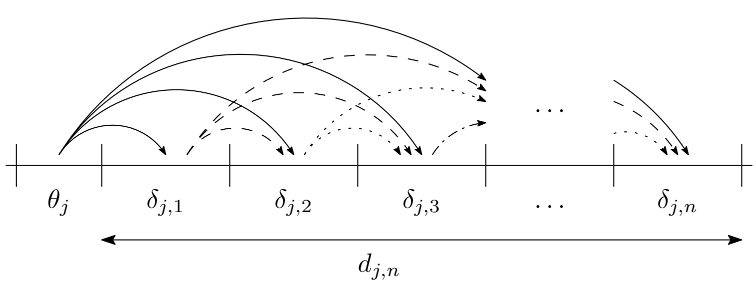


Figure 3: Description of $d_{j,n}$.

In addition, we denote by $d_{j,n}$ the total mean work in $Q_{j+1}, \ldots, Q_{j+n}$ which originate from customer orders that arrive per unit of pick time B_j or travel time S_{j-1} and all the subsequent picks that are triggered by these picks before the picker finishes service in Q_{j+n}. For example, a single product pick in Q_j will generate on average additional work in $Q_{j+1}, \ldots, Q_{j+n}$ of duration $E(B_j)\, d_{j,n}$. Then $d_{j,0} = 0$ and for $n > 0$ we have,

$$d_{j,n} = \sum_{m=1}^{n} \delta_{j,m}, \quad j = 1, \ldots, N, \tag{1}$$

where $\delta_{j,m}$ is the contribution of Q_{j+m}. First, $\delta_{j,1} = \rho_{j+1}/\left(1-\rho_{j+1}\right)$ includes the mean picking times and the consecutive busy periods in Q_{j+1} of product units that arrived during a product pick B_j or travel time S_{j-1}. Then, $\delta_{j,2} = \left(1+\delta_{j,1}\right)\rho_{j+2}/\left(1-\rho_{j+2}\right)$ contains the mean picking times of the product units that arrived in Q_{j+2} during B_j or S_{j-1} and the previous busy periods in Q_{j+1} plus all the busy periods that these picks generate in Q_{j+2}. In general we can write $\delta_{j,n}$ for $n > 0$ as (see Figure 3),

$$\delta_{j,n} = \sum_{m=1}^{\min(N-1,n)} \delta_{j,n-m} \frac{\rho_{j+n}}{1-\rho_{j+n}}, \quad j = 1, \ldots, N,$$

where $\delta_{j,0} = 1$. Note that $\delta_{j,n}$ only depends on at most $N-1$ previous $\delta_{j,n-m}$'s because if new demand arrives at the queue that is currently being visited it will be picked before the end of the current visit.

The mean order throughput time $E\left(T\right)$ can be determined by explicitly conditioning on the location of the order picker and by studying the system until the incoming customer order has been fully delivered at the depot,

$$E\left(T\right) = \frac{1}{E\left(C\right)} \sum_{j=1}^{N} E\left(\theta_j\right)\left(\pi\left(\mathcal{K}_j^0\right) E\left(T^{(\theta_j,0)}\right) + \pi\left(\mathcal{K}_j^1\right) E\left(T^{(\theta_j,1)}\right)\right). \tag{2}$$

Whenever the order picker is at intervisit period θ_j and still can pick all the products of an incoming customer order (i.e. $\boldsymbol{k} \in \mathcal{K}_j^0$), then the order throughput time is equal to $E\left(T^{(\theta_j,0)}\right)$. This is the mean time until the order picker reaches the depot during the current cycle including the conditional mean number of picks for customer orders in $\boldsymbol{k} \in \mathcal{K}_j^0$. Otherwise, one or more products are located upstream and the order throughput time is equal to $E\left(T^{(\theta_j,1)}\right)$. This is the expected time until the order picker reaches the depot in the next cycle including the conditional mean number of picks for customer orders in $\boldsymbol{k} \in \mathcal{K}_j^1$.

First, we focus on the derivation of $E\left(T^{(\theta_j,0)}\right)$. When the customer order enters the system in intervisit period θ_j with probabilities $E\left(V_j\right)/E\left(\theta_j\right)$ and $E\left(S_{j-1}\right)/E\left(\theta_j\right)$ it has to wait for a residual picking time $E\left(B_j^R\right) = E\left(B_j^2\right)/\left(2E\left(B_j\right)\right)$ or residual travel time $E\left(S_{j-1}^R\right) = E\left(S_{j-1}^2\right)/\left(2E\left(S_{j-1}\right)\right)$. Also, it has to wait for $E\left(\bar{L}_j^{(\theta_j)}\right)$ product units that still need to be picked at Q_j, as well as the expected $E\left(K_j|\mathcal{K}_j^0\right)$ product units that need be picked at this queue for a customer order in $\boldsymbol{k} \in \mathcal{K}_j^0$. Each of these picks triggers a busy period of length $E\left(B_j\right)/\left(1-\rho_j\right)$ and generates additional picks that will be made before the end of the current cycle of duration $d_{j,N-j}E\left(B_j\right)/\left(1-\rho_j\right)$. This also applies for the residual picking time and residual travel time. Then, for each subsequent intervisit period θ_l, $l = j+1, \ldots, N$, the travel time from Q_{l-1} to Q_l will trigger a busy period and additional picks in $Q_l, \ldots, Q_N$ of duration $E\left(S_{l-1}\right)\left(1+d_{l,N-l}\right)/\left(1-\rho_l\right)$. Similarly, the average number of product units that still needed to be picked at the customer order arrival and the mean $E\left(K_l|\mathcal{K}_j^0\right)$ product units needed to be picked for the arriving customer order will increase the order throughput time by $\left[E\left(\bar{L}_l^{(\theta_j)}\right)+E\left(K_l|\mathcal{K}_j^0\right)\right]E\left(B_l\right)\left(1+d_{l,N-l}\right)/\left(1-\rho_l\right)$. Finally, the picked orders have to be delivered to the depot of which the duration is $E\left(S_N\right)$.

Combining this gives the following expression for the mean time until the order picker reaches the depot during the current cycle given the average number of picks for a customer order in $\boldsymbol{k}\in\mathcal{K}_j^1$,

$$
\begin{aligned}
E\left(T^{(\theta_j,0)}\right) = & \big[\frac{E\left(V_j\right)}{E\left(\theta_j\right)}E\left(B_j^R\right)+\frac{E\left(S_{j-1}\right)}{E\left(\theta_j\right)}E\left(S_{j-1}^R\right)+\left[E\left(\bar{L}_j^{(\theta_j)}\right)+E\left(K_j|\mathcal{K}_j^0\right)\right]E\left(B_j\right)\big] \\
& \times\frac{1+d_{j,N-j}}{1-\rho_j}+\sum_{l=1}^{N-j}\Big(E\left(S_{j+l-1}\right)+\left[E\left(\bar{L}_{j+l}^{(\theta_j)}\right)+E\left(K_{j+l}|\mathcal{K}_j^0\right)\right]E\left(B_{j+l}\right)\Big) \\
& \times\frac{1+d_{j+l,N-j-l}}{1-\rho_{j+l}}+E\left(S_N\right). \quad (3)
\end{aligned}
$$

Next we focus on $E\left(T^{(\theta_j,1)}\right)$. The derivation is similar to the one of Equation (3), except that we should also consider the additional demand that is generated during a pick or a switch from queue to queue until the end of the next picking cycle. This gives the following expression,

$$
\begin{aligned}
E\left(T^{(\theta_j,1)}\right) = & \big[\frac{E\left(V_j\right)}{E\left(\theta_j\right)}E\left(B_j^R\right)+\frac{E\left(S_{j-1}\right)}{E\left(\theta_j\right)}E\left(S_{j-1}^R\right)+\left[E\left(\bar{L}_j^{(\theta_j)}\right)+E\left(K_j|\mathcal{K}_j^1\right)\right]E\left(B_j\right)\big] \\
& \times\frac{1+d_{j,2N-j}}{1-\rho_j}+\sum_{l=1}^{N-1}\Big(E\left(S_{j+l-1}\right)+\left[E\left(\bar{L}_{j+l}^{(\theta_j)}\right)+E\left(K_{j+l}|\mathcal{K}_j^1\right)\right]E\left(B_{j+l}\right)\Big) \\
& \times\frac{1+d_{j+l,2N-j-l}}{1-\rho_{j+l}}+\sum_{l=N}^{2N-j}E\left(S_{j+l-1}\right)\frac{1+d_{j+l,2N-j-l}}{1-\rho_{j+l}}+E\left(S_N\right). \quad (4)
\end{aligned}
$$

Then, $E\left(T\right)$ in (2) can be easily calculated with use of (3) and (4).

4 Numerical results

In this section we investigate the mean order throughput time for a real world milkrun picking system. In addition, we also study the effects of different product allocations. For this we study the warehouse of an online Chinese retailer in consumer electronics, the same warehouse considered in case 2 in Gong & De Koster [3]. However, the authors only compared the product unit waiting times. The retailer sells over $20,000$ products in 226 cities and provides deliveries within 2 hours upon order receipt in large cities. In order to meet this service level agreement, management requires that orders should start processing within 5 minutes on average after being received and the order throughput times should be as short as possible.

Table 1: Parameters of the China online shopping warehouse.

(a) Warehouse

Parameter	Value
Warehouse area	985 m^2
Aisles	8
Number of storage locations per aisle side	30

(b) Order pickers

Parameter	Value
Number of order pickers	30
Number of storage locations per picker, N	16
Number of aisles per picker, A	4
Number of storage locations per rack per picker, L	2

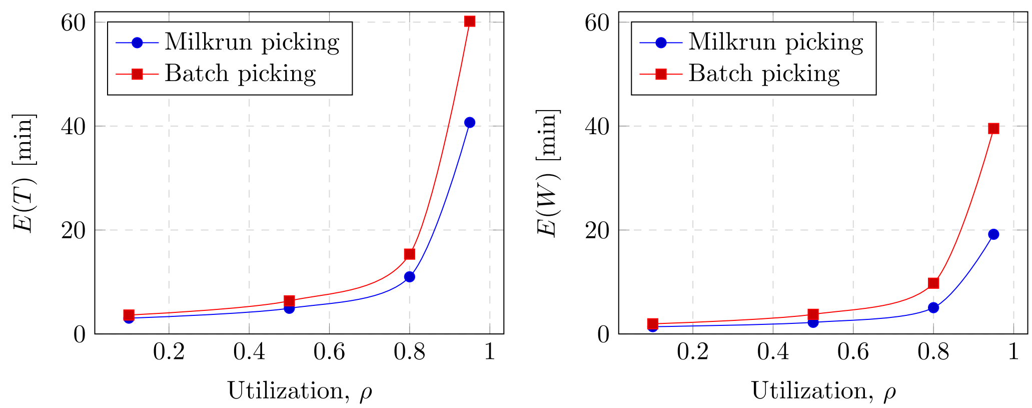
(c) Operations

Parameter	Value
Travel speed of a picker	0.48 meter/sec.
Mean picking time, $E\left(B_i\right)$	1.51 sec.
Second moment picking time, $E\left(B_i^2\right)$	3.82
Mean traveling time (depot), s_0	63.0 sec.
Mean traveling time (side to side), s_1	2.00 sec.
Mean traveling time (adjacent storage locations), s_2	2.50 sec.
Mean traveling time (adjacent aisles), s_3	9.60 sec.

The company uses a milkrun picking system aided by an information system based on mobile technology and a call center (order processing center). In Table 1 an overview of the parameters of the warehouse is provided. The total area dedicated for the milkrun picking system is $985\ m^2$. The total number of aisles is 8 and each aisle has a width of 1 meter. On each side of the aisle there are 30 storage positions, where each storage position has a width and depth of 1.2 meter. Altogether there are $480\ (= 2 \cdot 8 \cdot 30)$ storage locations.

In total there are now 30 order pickers working per shift in the warehouse. Different from Gong & De Koster [3] who assume all order pickers visit sequentially every storage location and thus follow the same picking route, we assume that the order picking area is zoned and each picker is responsible for picking products from his or her zone. This means that there is no overlap in picking routes between order pickers. Picked products are brought to a central depot location where they are sorted per customer order. Additionally, we assume 64% of the incoming orders request only one product unit and 36% two product units. Also, we assume that every customer order can be fully picked in one zone since they contain distinct enough product ranges. This allows us to study each zone in isolation.

Then, a single order picker is responsible for $N = 16 = 2 \cdot 4 \cdot 2$ storage locations. The subsequent picking routes can be realized by adding additional cross-aisles to the order picking area. Each order picker has a traveling speed of 0.48 meter/seconds. The mean travel time side to side is $s_1 = 2$ seconds, the mean travel time within aisles between adjacent storage location is $s_2 = 2.50$ seconds, and the mean travel time between adjacent aisles is $s_3 = 9.60$ seconds. The average mean traveling times from the last storage location including the depot time is $s_0 = 63.0$ seconds for all the pickers. As a result, the total mean traveling time per cycle is $E(S) = 182.2$ seconds. All the second moments for the traveling times are $s_i^2 = 0$, $i = 0, 1, 2, 3$. Finally, for all storage locations the mean picking time per product unit is $E(B_i) = 1.51$ seconds and second moment of the picking time is $E(B_i^2) = 3.82$, $i = 1, \ldots, N$. In the rest of this section, we focus on one zone but the same conclusion can also be drawn for the other zones. All the experiments were run on Core i7 with 2.5 GHz and 8 GB of RAM.



(a) Mean order throughput time $E(T)$ (b) Mean product unit waiting time $E(W)$

Figure 4: Results China online shopping warehouse for different utilization ρ.

Figure 4 shows the mean order throughput time and mean product unit waiting time in case of milkrun picking and conventional batch picking. For both systems and each ρ we generated a large set of different product allocation in order to find the best allocation x. For the batch picking situation we assume that the order picker has to visit all the picking locations during a picking tour and immediately goes on a next tour after delivering the products at the depot and that no orders can be included during the current picking cycle. Therefore, we can analyze this situation with the results in case of globally-gated from Van der Gaast et al. [6]. In Figure 4a the results for the mean order throughput time $E(T)$ is shown. The milkrun picking always achieves the lowest mean order throughput time and the conventional batch picking performs significantly worse which shows that dynamically adding new customer orders to the picking cycle reduces the mean order throughput times considerably. The number of product units ranges from a few when the utilization is low to higher than 50 in case utilization is high. From the results it can also be clearly seen that when the utilization increases, the mean order throughput times increases rapidly. For the average mean product unit waiting time $E(W) = \frac{1}{\Lambda}\sum_{i=1}^{N} \lambda_i E(W_i)$ in Figure 4b, similar conclusions can be drawn. On the other hand, comparing the results with the mean order throughput time it can be seen that the mean order throughput time is between 50% to 125% longer. This implies that when considering how long it takes to pick a customer order it is better to consider the order throughput time instead of product unit waiting time.

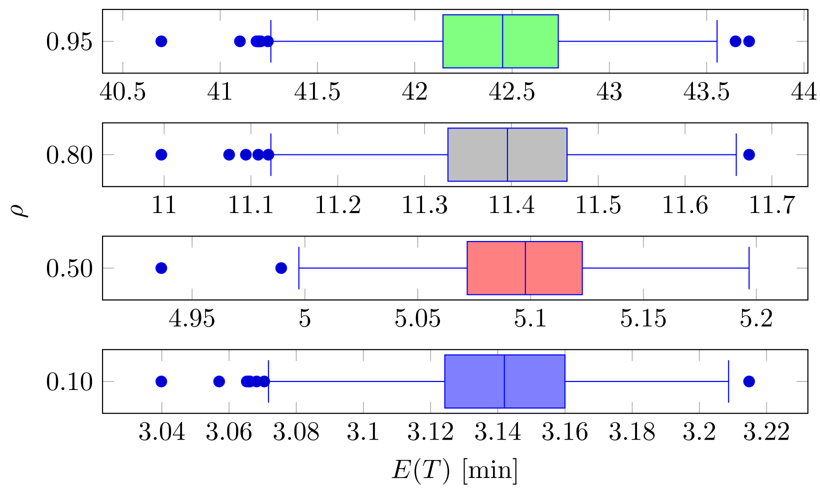


Figure 5: Box plots of mean order throughput times for $3,000$ different product allocations for various values of the utilization ρ.

Next, Figure 5 shows how much the mean order throughput time varies for several values of the utilization ρ for the randomly generated set of product allocations x. We generated $3,000$ different allocations which also included the best allocation found in Figure 4 for which we calculated the mean order throughput time $E(T)$. From the box plots it can been seen that the spread of mean order throughput times is around 3 minutes in case ρ is low to a couple of seconds when ρ is high and that a proper picking strategy can lead to significantly shorter order throughput times. In conclusion, from the results it can be seen that the best storage allocation can improve the order throughput time around 10% compared to the worst storage allocation.

5 Conclusion and further research

This paper studied the mean order throughput time in a milkrun picking system. This allows us to gain a better insights in the performance of the system and allows to study the effect of different product allocations. For a real world application we show that milkrun order picking reduces the order throughput time significantly in case of high arrival rates compared to conventional batch picking within the set of modelling assumptions. In addition, the best storage allocation can improve the order throughput time around 10% compared to the worst storage allocation.

The model and methods in this paper lend themselves for further research. First, the model can be extended by including putaway and replenishment processes, similar as observed in a production setting. Other interesting topics are relaxing the assumption of an uncapacitated pick cart and investigating whether other or combinations of picking strategies can lead to increased picking performance. Also, it can be worthwhile to investigate whether a local backward routing policy, i.e. picking a product that arrived in the queue that just has been visited, might increase system performance. Another research topic is to analyze a multi-zoned warehouse where each zone operates under a separate milkrun.

References

[1] De Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2), 481–501.

[2] Gademann, N. & Van de Velde, S. (2005). Order batching to minimize total travel time in a parallel-aisle warehouse. *IIE transactions*, 37(1), 63–75.

[3] Gong, Y. & De Koster, R. (2008). A polling-based dynamic order picking system for online retailers. *IIE Transactions*, 40(11), 1070–1082.

[4] Gong, Y. & De Koster, R. B. M. (2011). A review on stochastic models and analysis of warehouse operations. *Logistics Research*, 3(4), 191–205.

[5] Takagi, H. (1986). *Analysis of polling systems*. MIT press.

[6] Van der Gaast, J. P., Adan, I. J. B. F., & de Koster, R. B. M. (2015). The analysis of batch sojourn-times in polling systems. Working paper.