2014

Modeling Conveyor Merges in Zone Picking Systems

Rene De Koster
Technology and Operations Management, rkoster@rsm.nl

Ivo Adan
Technische Universiteit Eindhoven, iadan@win.tue.nl

Jacques Resing
Technische Universiteit Eindhoven, resing@win.tue.nl

XXXI.
Modeling conveyor merges in zone picking systems

Jelmer van der Gaast, René de Koster
Technology and Operations Management,
Erasmus Universiteit Rotterdam, The Netherlands

Ivo Adan
Department of Mechanical Engineering,
Technische Universiteit Eindhoven, The Netherlands

Jacques Resing
Department of Mathematics and Computer Science,
Technische Universiteit Eindhoven, The Netherlands

Abstract

In many order picking and sorting systems conveyors are used to transport products through the system and to merge multiple flows of products into one single flow. In practice, conveyor merges are potential points of congestion, and consequently can lead to a reduced throughput. In this paper, we study merges in a zone picking system. The performance of a zone picking system is, for a large part, determined by the performance of the merge locations. We model the system as a closed queueing network that describes the conveyor, the pick zones, and the merge locations. The resulting model does not have a product-form stationary queue-length distribution. This makes exact analysis practically infeasible. Therefore, we approximate the behavior of the model using the aggregation technique, where the resulting subnetworks are solved using matrix-geometric methods. We show that the approximation model allows us to determine very accurate estimates of the throughput when compared with simulation. Furthermore, our model is in particular well suited to evaluate many design alternatives, in terms of number of zones, zone buffer lengths, and maximum number of totes in the systems. It also can be used to determine the maximum throughput capability of the system and, if needed, modify the system in order to meet target performance levels.

1 Introduction

Conveyor systems are a critical component of many order picking and sorting systems in that they move products from one location to another. One of the most important uses of conveyor systems is to consolidate multiple flows of products into one single flow (a *merge* operation). These merges are potential points of congestion which can lead to blocking and increased order throughput times. Obviously, the performance of the merges strongly influences the performance of the overall system.

A very popular order picking method in practice that uses conveyors is *zone picking*. Zone picking is a picker-to-parts order picking method, which divides the order picking area in work zones, each operated by one or multiple order pickers [1, 2]. Orders visit sequentially the work zones until they are completed. Zone picking systems often use a conveyor to transport orders between the zones. Major advantages of zone picking systems are the high-throughput ability, scalability, flexibility in handling both small and large order volumes, and handling different product sizes, with a different number of order pickers. These systems are often applied in e-commerce warehouses handling customer orders with a large number of order lines and with a large number of different products kept in stock [3]. A disadvantage, however, is that under heavy load, congestion and blocking can occur due to finite zone buffers and conveyor merges. This congestion leads to a reduced throughput and causes unpredictable throughput times. As a direct consequence, orders cannot be shipped on time which leads to delayed customer deliveries and loss in revenue. Especially e-commerce warehouse companies deal with very strict delivery lead times since its customers demand fast delivery, often within 24 hours.

Next to delivery lead times, *throughput* is a key performance indicator in zone picking systems. Throughput, measured as the number of completed orders/order lines per period of time, is used to judge whether the order picking system is capable to perform according to a certain customer demand. However, estimating the throughput of a zone picking system, or any conveyor system in general is very complicated, especially in the presence of finite buffers, intermediate storage, job variability, and variability of the performance of components and human operators.

Often, zone picking systems are analyzed by developing simulation models and testing various scenarios. Simulation can allow for very accurate modeling, but it can be expensive to build the simulation model and the time needed to evaluate each scenario or lay-out design can be significant, especially when the system is highly utilized and blocking can occur. Also, the accuracy of the simulation is strongly dependent on the quality of the calibration data [4]. Another approach to analyze zone picking systems are queueing networks. Queueing networks are in general much faster, more flexible, and less data expensive in determining the performance of a zone picking system. They can subsequently be used as good evaluation tools in the initial design phase in order to help designers quickly evaluate many design alternatives and narrow down the available design choices [2]. They can even be used to optimize the system in later phases in terms of order release rules and workload allocation.

The objective of this paper is to quantify the impact of merge operations on the throughput of zone picking systems. The system is modeled as a closed queueing network that describes the conveyor, the pick zones and the merge locations. The resulting model does not have a product-form stationary queue-length distribution which makes exact analysis practically infeasible. Therefore, we approximate the behavior of the model using the aggregation technique [5], where the resulting subnetworks are solved using matrix-geometric methods [6]. In addition, we also study the merge operations in case of the dynamic block-and-recirculate protocol [7] that is often encountered in zone picking systems. We show that the approximation model allows us to determine very accurate estimates of the throughput of zone picking system with merge operations when compared with simulation. The model is in particular well suited to evaluate many design alternatives, in terms of number of zones, zone buffer lengths, and maximum number of totes in the systems, and can be used to determine the maximum throughput capability of the system and, if needed, modify the system in order to meet target performance levels.

The organization of this paper is as follows. In Section 2 zone picking systems are discussed. An overview of existing models for both zone picking and conveyor systems with recirculating loops and merge operations is given in Section 3. The queueing model is presented in Section 4. In Section 5 our approximation method is explained and it is verified for its performance in Section 6 via computational experiments for a range of parameters. In the final section we conclude and suggest some extensions of the model.

2 Zone picking systems

In zone picking systems the order picking area is zoned so that each order picker is responsible for picking products only from his/her zone. Zone picking systems can be categorized in either *parallel* or *sequential* zone picking [8].

In a parallel zone picking system, multiple pickers, in multiple zones, can work simultaneously on one order (or a batch of orders). The picked products are sent downstream to a designated consolidation area where they are combined into orders. On the other hand, in sequential zone picking (or pick-and-pass), an order is assigned to an order tote or order carton that travels on the conveyor and is directed sequentially to the next zone where products are stored that should be added to the order. At a zone, each picker picks for only one tote at a time. The advantage of sequential zone picking is that order integrity is maintained and no sorting and product consolidation is required [9]. Such zone picking systems are highly popular in practice, especially in case of e-commerce warehouses, that have many outstanding orders to be picked. In this paper we will only consider sequential zone picking (hereafter zone picking).

In Figure 1 a schematic representation is shown of a zone picking system where the picking area is divided into two zones. These zones are connected by conveyors enabling

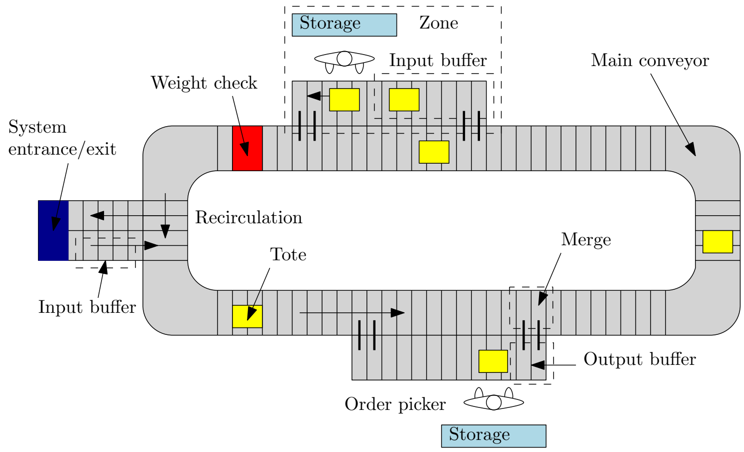


Figure 1: A zone picking system with a single-segment and two zones

automatic transportation of customer orders through the system. A customer order is released at the system entrance as an order tote, which contains a list of products to be picked and their locations within the picking area. The tote only enters the system when it is allowed by the workload control mechanism [3]. This mechanism sets an upper bound on the number of totes in the system and only releases a new tote when a tote with all required order lines leaves the system. After release, the tote travels on the main conveyor to the required zones and enters the input buffer when reaching one of these zones where it waits until it is first in the queue. The order picker then starts picking the required products that are located within the zone. After all picks are completed the order picker places the tote in the output buffer, where the tote waits until there is enough space on the main conveyor such that it can travel to a next zone. When the tote has visited all the required zones, it leaves the system at the exit and a new waiting order tote can be released into the system.

Zone picking systems can differ in many respects, e.g. workstation type, pick-face, buffer lengths, storage size, and conveyor configuration. Especially the conveyor configuration is of great importance, since it affects how and when totes arrive at the zones. In many zone picking systems, a tote can skip a zone in case it does not require products to be picked from the zone. Also, combined with a closed-loop conveyor, totes can skip the zone when the zone's input buffer is fully occupied. The tote can return to this zone after visiting potentially other zones or after it recirculated on the conveyor (a weight check at the end of conveyor ensures that the tote will not leave system before visiting all the required zones). The advantage of this dynamic block-and-recirculate protocol is that it prevents congestion on the main conveyor and balances workload across the various zones. For a detailed analysis of this protocol in zone picking the reader is referred to van der Gaast et al. [7]. Note that this blocking protocol shares similarities with the blocking protocol encountered in closed-loop flexible manufacturing systems [10]. The key difference is, however, that in zone picking totes can visit the required zones in a random order and they do not need to return immediately to the same blocked zones, whereas in flexible manufacturing systems the current task has to be performed on either the blocked machine or on a complementary identical machine before starting the next task.

The performance of a zone picking system is, for a large part, determined by the performance of the merges. After entering the system and after each zone, totes should merge on the main conveyor in order to move to their next location. At a merge totes already on the main conveyor have absolute priority (as this conveyor belt has no possibilities for accumulation). Therefore, in order to allow a tote to leave the output buffer, its predecessors from the same buffer should have left while, in addition, a sufficiently large gap on the main conveyor must be present to prevent collisions. Typically, the time required to create a sufficiently large space on the conveyor is negligible. However, in highly utilized systems, this space can become very scarce leading to long merge times and a loss in overall performance of the system. In addition, the output buffer can become full and stop the order picker (or the entrance station) from continuing to work on the next tote in line. Only when there is at least one empty place in the output buffer, the order picker/entrance station can resume his or her work. Finally, in some zone picking systems there is no output buffer. In these cases the order picker/entrance station must always wait until the tote has entered the main conveyor before starting to work on the next tote in line.

3 Existing literature

Literature on zoning picking systems is still very limited, although the subject has started to gain popularity in recent years. Gray et al. [11] used a hierarchical approach to evaluate economic tradeoffs of equipment selection, storage assignment, number of zones, picker routing, and order batching when designing a zone picking system. De Koster [12] modeled a zone picking system without recirculation as a Jackson queueing network which allows for fast early-stage estimation of design alternatives in terms of order throughput times and average work-in-process. Malmborg [13] developed a model to study the tradeoffs in space requirements and retrieval costs with dedicated and randomized storage in a zone picking system. Workload balancing in zone picking systems is studied by Jane [14], who proposed several heuristic methods to adjust the number of zones so that all pickers stay balanced. Petersen [1] performed a simulation study to investigate the shape of the zone and showed that the size or storage capacity of a zone, the number of items on the pick list, and the storage policy have a significant effect on the average walking distances. Jewkes et al. [15] studied the assignment of products to zones and the location of the picker home base in order to minimize the expected order cycle time. For fixed product locations, the authors developed a dynamic programming algorithm that optimally determines the product and server locations. Yu and De Koster [16] analyzed zone picking systems without recirculation and presented an approximation method based on a $G/G/m$ queueing network. Eisenstein [17] analyzed product assignments and depot locations in a zone picking system when single or dual depots are allowed along the pick line. Pan and Wu [18] used Markov chain analysis and proposed three heuristics that optimally allocate items to a single picking zone, a picking line with unequal-sized zones, and a picking line with equal-sized zones. Melacini et al. [19] modeled a zone picking system as a network of queues. In order to estimate performance statistics, such as the utilization, throughput rate of a zone, and the mean and standard deviation of the throughput time of the totes, they use Whitt's queueing network analyzer [20].

In contrast, the analysis of conveyor systems has received much more attention. The models from the literature can categorized as either deterministic or stochastic. Deterministic conveyors models were studied by e.g. Kwo [21], Muth [22], Bastani and Elsayed [23], Bastani [24] who investigate feasibility conditions, such as loading/unloading rates and conveyor lengths, for various simple closed-loop conveyors. However, these models fail to capture the effects that random fluctuations in either the input or output can have on the design and performance of conveyor systems.

Stochastic models have been studies by various authors. Disney [25] studied the behavior of a conveyor system as a multichannel queueing system with ordered entry. This model served as the basis for many other studies about conveyor systems (see Muth and White [26] for a survey of these models). Sonderman [27] studied a conveyor system with a single loading and unloading station where loads can recirculate. The author uses Whitt's queueing network analyzer [20] to approximate the output process at the unloading station and to study the effect of recirculation. Sonderman and Pourbabai [28] extend the model of Sonderman [27] by allowing random access on the conveyor.

Coffman Jr et al. [29] studied a conveyor system for flexible manufacturing systems and investigated the effect of the distance of input and output points of workstations at which items leave and rejoin the conveyor. In order to study the performance of the system, the conveyor queue was modeled by a Markov process. Schmidt and Jackman [30] modeled a recirculating conveyor as an open network of queues. The system consists of one loading station, one unloading station, and two servers performing the same service on loads entering the system, and a loop conveyor divided into segments. Zijm et al. [31] analyzed an automated kit transportation system and studied a number of key elements of the system separately and subsequently combined the results of this analysis in an Approximate Mean Value Analysis (AMVA) algorithm. Bozer and Hsieh [32] and Hsieh and Bozer [33] modeled a conveyor as a unidirectional closed loop consisting of discrete spaces or windows of equal size, which hold at most one load or unit. They consider different machines that are located around the conveyor with a pair of unloading and loading stations per machine, modeling them as output and input queues.

All these papers analyze only particular aspects that are relevant for a zone picking system, such as recirculation and merging conveyor flows. In the next section, we integrate these various aspects into a one single model.

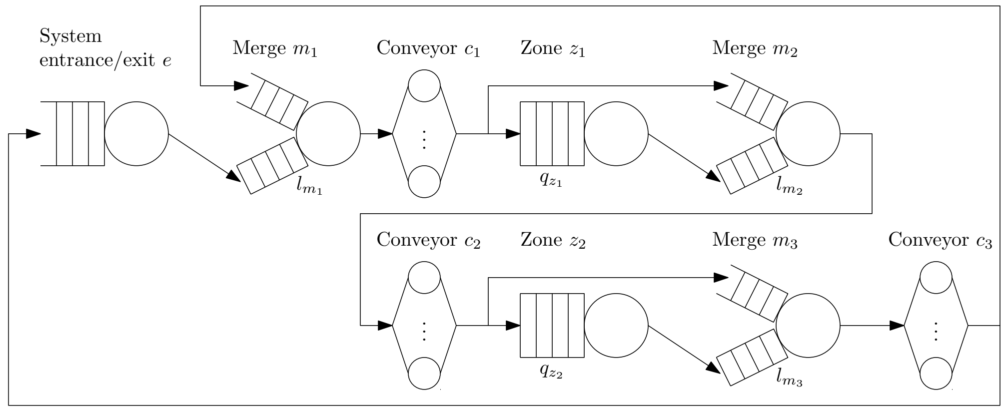


Figure 2: The corresponding queueing network with system entrance/exit station e, conveyors $\mathcal{C} = \{c_1, c_2, c_3\}$, merges $\mathcal{M} = \{m_1, m_2, m_3\}$ and, zones $\mathcal{Z} = \{z_1, z_2\}$

4 Queueing model for zone picking systems

The model for zone picking systems with merges is shown in Figure 2 for the case case of two zones. van der Gaast et al. [7] studies a similar model containing zones and conveyors, but they did not explicitly model the merges. The zone picking system is modeled as a closed queueing network with one entrance/exit, W zones, $W + 1$ merges, and $W + 1$ nodes that describe the conveyor between a merge location and a zone or the entrance/exit. The nodes are labeled in the following manner: the system entrance/exit is denoted as e, $\mathcal{Z} = \{z_1, \ldots, z_W\}$ denotes the set of zones, $\mathcal{M} = \{m_1, \ldots, m_{W+1}\}$ denotes the set of merges, and $\mathcal{C} = \{c_1, \ldots, c_{W+1}\}$ is the set of conveyors in the network. Finally, let $\mathcal{S} = \{e\} \cup \mathcal{C} \cup \mathcal{M} \cup \mathcal{Z}$ be the union of all the nodes in the network. The following assumptions are adopted for the network:

- There is an infinite supply of totes at the entrance of the system. This means that a leaving tote can always be replaced immediately by a new tote. Each tote has a class $\boldsymbol{r} \subseteq \mathcal{Z}$, e.g., $\boldsymbol{r} = \{z_2, z_3\}$ means that the tote has to visit the second and third zone.

- The total number of totes in the system is constant N. As long as the total number of totes in the zones, merges, and conveyor nodes is less than N, new totes are released one-by-one at an exponential rate μ_e at the system entrance. The assumption of constant N totes in the system is not restrictive, because most zone picking systems, especially in an e-commerce warehouse, are typically heavily utilized. This means that at any point in time there are always a sufficient outstanding customer orders needed to be picked.

- The conveyor nodes are assumed to be infinite-server nodes with a deterministic delay of rate μ_i, $i \in \mathcal{C}$.

- Each zone has only one order picker. The order picking time is assumed to be exponentially distributed with rate μ_i, $i \in \mathcal{Z}$, that captures both variations in the pick time per tote and variations in the number of order lines to be picked.

- When the order picker is busy, incoming totes are stored in a finite input buffer of size $q_i\,(\geq 0)$, $i \in \mathcal{Z}$. Incoming totes are blocked when the total number of totes in the input buffer equals q_i.

- The merge nodes are assumed to be single server preemptive-repeat (different) priority stations where totes on the main conveyor (*high priority*) have absolute priority over the totes flowing out of the zones/entrance (*low priority*). Whenever a high priority tote enters the merge, it will preempt any low priority tote currently in service. After the high priority tote has left and no other tote of high priority is currently at the merge, the low priority tote will repeat its service. The time required to pass the merge, either for low or high priority totes, is assumed to be exponentially distributed with rate μ_i, $i \in \mathcal{M}$.

- Each merge has a limited capacity of size $l_i\,(\geq 0)$, $i \in \mathcal{M}$ to store low priority totes. This corresponds with the limited output buffer found after the zones/entrance. When there are l_i low priority totes waiting at the merge node, no incoming low priority tote will be accepted by the merge node and it has to wait at its current node, subsequently blocking the order picker/entrance station from starting to work on the next tote in line. For the unblocking procedure we can distinguish two distinct cases. In case $l_i \geq 1$, only when there is at least one open position for low priority totes at the merge node, the tote leaves its current node and unblocks the order picker/entrance station. Whenever there is no output buffer ($l_i = 0$), the order picker/entrance station only unblocks after the tote has passed the merge.

Let $\mathbb{S}\,(N)$ be the state space of the network, i.e., the set of states $\boldsymbol{x} = (\boldsymbol{x}_i : i \in \mathcal{S})$ for which the number of totes in the system is equal to $\sum_{i\in\mathcal{S}} n_i = N$, where n_i is the number of totes in node i. The state of conveyor $i \in \mathcal{C}$ is $\boldsymbol{x}_i = (\boldsymbol{r}_{i1}, \ldots, \boldsymbol{r}_{in_i})$ with $\boldsymbol{r}_{i1}$ as the class of the first tote in the node, and $\boldsymbol{r}_{in_i}$ as the class of the last tote in the node. The state of zone $i \in \mathcal{Z}$ and entrance/exit e is defined similarly, except that it only includes the totes waiting in the input buffer and the totes currently receiving service. Let $0 \leq t_i \leq 1$, $i \in \{e\} \cup \mathcal{Z}$ be the number of totes that have received service at the zone/entrance but are still waiting at this node since they cannot enter the output buffer or cross the merge. These totes are placed at the back of the low priority totes in the state of the next merge $i \in \mathcal{M}$ in line, which is defined as $\boldsymbol{x}_i = \left(\boldsymbol{r}^H_{i1}, \ldots, \boldsymbol{r}^H_{in^H_i}; \boldsymbol{r}^L_{i1}, \ldots, \boldsymbol{r}^L_{in^L_i}\right)$ with $n_i = n^H_i + n^L_i$ as the number of totes with high and low priority respectively. In addition, the number of totes in each zone satisfies the capacity constraint $n_i + t_i \leq q_i + 1$, $i \in \mathcal{Z}$, which states that a tote cannot enter the zone if the input buffer is full and the order picker is occupied/blocked. Finally, the number of low priority totes in each merge should not exceed the capacity of the input buffer l_i and the single server of the merge; $n_i^L \leq l_i + 1$, $i \in \mathcal{M}$.

Denote by $p_{i\boldsymbol{r},j\boldsymbol{s}}(\boldsymbol{x})$ the state dependent routing probability that a tote of class $\boldsymbol{r}$ is routed from node i to node j and enters as a class $\boldsymbol{s}$ tote given that the network is in state $\boldsymbol{x}$. The routing probabilities of the network can be written as follows:

$$p_{e\boldsymbol{s},m_1\boldsymbol{r}}(\boldsymbol{x}) = \psi_{\boldsymbol{r}}, \tag{1}$$
$$p_{m_i\boldsymbol{r},c_i\boldsymbol{r}}(\boldsymbol{x}) = 1, \quad i = 1, \ldots, W, \tag{2}$$
$$p_{c_i\boldsymbol{r},z_i\boldsymbol{r}}(\boldsymbol{x}) = 1, \quad i = 1, \ldots, W,\ z_i \in \boldsymbol{r} \text{ and } n_{z_i} + t_{z_i} < q_{z_i} + 1, \tag{3}$$
$$p_{c_i\boldsymbol{r},m_{i+1}\boldsymbol{r}}(\boldsymbol{x}) = 1, \quad i = 1, \ldots, W,\ z_i \notin \boldsymbol{r} \text{ or } n_{z_i} + t_{z_i} = q_{z_i} + 1, \tag{4}$$
$$p_{z_i\boldsymbol{r},m_{i+1}\boldsymbol{s}}(\boldsymbol{x}) = 1, \quad i = 1, \ldots, W,\ \boldsymbol{s} = \boldsymbol{r} \backslash \{z_i\}, \tag{5}$$
$$p_{c_{W+1}\boldsymbol{r},e\boldsymbol{r}}(\boldsymbol{x}) = 1, \quad \boldsymbol{r} = \emptyset, \tag{6}$$
$$p_{c_{W+1}\boldsymbol{r},m_1\boldsymbol{r}}(\boldsymbol{x}) = 1, \quad \boldsymbol{r} \neq \emptyset. \tag{7}$$

Every other probability equals to 0.

A new tote of class $\boldsymbol{r} \subseteq \mathcal{Z}$ is released at the system entrance with probability $\psi_{\boldsymbol{r}}$. These release probabilities correspond to a known order profile that can be obtained using e.g., historical order data or forecasts. After release, a tote of class $\boldsymbol{r}$ moves from the system entrance to the first merge node m_1 (1). In general after merging, a tote travels to conveyor node c_i (2). At conveyor node c_i, the tote will either enter the input buffer of zone i if $z_i \in \boldsymbol{r}$ and the buffer is not full (3) or move to the next merge m_{i+1} (4). In case the tote needs to enter and the buffer is full, the tote skips the zone and also moves to the next merge m_{i+1}, while it keeps the same class (4). If the buffer is not full, the tote enters the buffer of the zone and, after possibly waiting sometime in the buffer, the order picker picks the required order lines. After all picks are completed at zone z_i, the tote will enter the merge node and changes its class to $\boldsymbol{s} = \boldsymbol{r} \backslash \{z_i\}$ (5). In the merge node, the tote will wait before its predecessors have passed the merge point before it can start its service. Whenever during its service a tote from the conveyor tries to enter the merge, the tote is immediately preempted and waits until the merge becomes free in order to retry its service. When the tote successfully passes the merge it is routed to conveyor node c_{i+1}. After visiting the last conveyor node c_{W+1}, all the totes with $\boldsymbol{r} \neq \emptyset$ are routed to the first merge node m_1 (7); the other totes move to the exit and are immediately replaced by a new tote which is waiting for release at the entrance (6).

Exact analytic methods to analyze queueing networks are only known for a very limited set of models that satisfy certain conditions. The majority of these models have a *product-form* stationary distribution [34, 35, 36]. For these models, it can be proven that the stationary distribution of the network can be expressed as a product of factors describing the state of each node. Based on this independence assumption, exact efficient analysis algorithms such as the convolution algorithm [37] and the mean-value analysis (MVA) [38] can be applied to analyze the models.

However, the previously described queueing network does not have a product-form stationary distribution, because of the priorities at the merge nodes [39], and due to the dynamic block-and-recirculate protocol [7]. Also, direct analysis of the resulting underlying Markov chain is not feasible due to state-space explosion which excludes analyzing the Markov chain within reasonable time and storage. Usually, non-product-form queueing networks are studied using approximation analysis. An overview of many general techniques is presented in Bolch et al. [40].

In van der Gaast et al. [7] it is shown that the queueing network without merges is very accurately approximated by a related product-form queueing network with the *jump-over* protocol. The idea of the approximation is to replace the state dependent routing with state independent routing in such a way that the flows in the new network matches the flows of the original network. This is done by introducing a Bernoulli process that determines for every tote that intends to visit z_i, randomly, and independent of whether the tote actually visited z_i or not, whether the tote should return to z_i. The probability of the Bernoulli process b_{z_i} that a tote should return to a zone z_i, $i \in \mathcal{Z}$ is chosen in such a way that it corresponds with the probability that a tote is blocked by a zone in the original network. Naturally, blocking probabilities are not known in advance, but they are estimated iteratively after an initial guess from the approximation.

The queueing network with merges and the dynamic block-and-recirculate protocol can be transformed into a queueing network with jump-over blocking as follows. First of all, routing probabilities (1)-(7) become state independent. This means, in particular, after service at c_i each tote with $z_i \in \boldsymbol{r}$ is routed to z_i irregardless whether the buffer of the zone is full (8)-(9). The tote will enter the buffer if it is not full, otherwise the tote instantaneously skips the node. Then for each class $\boldsymbol{r}$ tote, independent of whether the tote visited or skipped z_i (because of a full buffer), $p_{z_i\boldsymbol{r},m_{i+1}\boldsymbol{s}} = 1 - b_{z_i}$, $i = 1, \ldots, M$, where $\boldsymbol{s} = \boldsymbol{r} \setminus \{z_i\}$. This means that a tote of class $\boldsymbol{r}$ is tagged *skipped* z_i and routed to the next conveyor node c_{i+1} with the same class with probability b_{z_i}, and otherwise, with probability $1 - b_{z_i}$, the tote is tagged as *visited* z_i and the class of the tote changes to $\boldsymbol{s} = \boldsymbol{r} \setminus \{z_i\}$.

$$p_{c_i\boldsymbol{r},z_i\boldsymbol{r}} = 1, \qquad i = 1, \ldots, W,\ z_i \in \boldsymbol{r}, \tag{8}$$

$$p_{c_i\boldsymbol{r},m_{i+1}\boldsymbol{r}} = 1, \qquad i = 1, \ldots, W,\ z_i \notin \boldsymbol{r}, \tag{9}$$

$$p_{z_i\boldsymbol{r},m_{i+1}\boldsymbol{r}} = b_{z_i}, \qquad i = 1, \ldots, W, \tag{10}$$

$$p_{z_i\boldsymbol{r},m_{i+1}\boldsymbol{s}} = 1 - b_{z_i}, \quad i = 1, \ldots, W,\ \boldsymbol{s} = \boldsymbol{r} \setminus \{z_i\}\,. \tag{11}$$

Since the recirculation process is made independent of the state of the buffer, essentially the block-and-recirculate protocol is replaced by the *jump-over* blocking protocol [41]. Under this protocol, each tote of class $\boldsymbol{r}$ leaving z_i, either after service or skipping, continues to follow the same Markovian routing. The advantage of the jump-over blocking protocol, also known as “overtake full stations, skipping, and blocking and rerouting", is that closed-form analytic results for single-class queueing networks are available in the literature [42, 43, 41, 44].

However, the jump-over network still has no product-form due to the finite capacity priority queues. In this paper, a decomposition-based approximation is developed by studying each merge location in isolation. We do not intend to investigate application of decomposition-based approximations to the network considered in this paper. Instead we focus on an alternative method that is feasible in terms of computational complexity and of very high accuracy as well. Our method progressively aggregates part of the network and replaces the aggregated subnetwork by a flow equivalent single node. To specify the state-dependent service rates of the flow equivalent node, we assume (approximately) independence between the subnetwork to be aggregated and the rest. Our approximation directly solves the global balance equation of the underlying Markov chain for its steady-state distribution. We are interested to see the quality of our method which follows the line of this well-known idea of aggregation, in the environment of non-product form networks, particularly for the network we consider here.

5 Approximate aggregation method

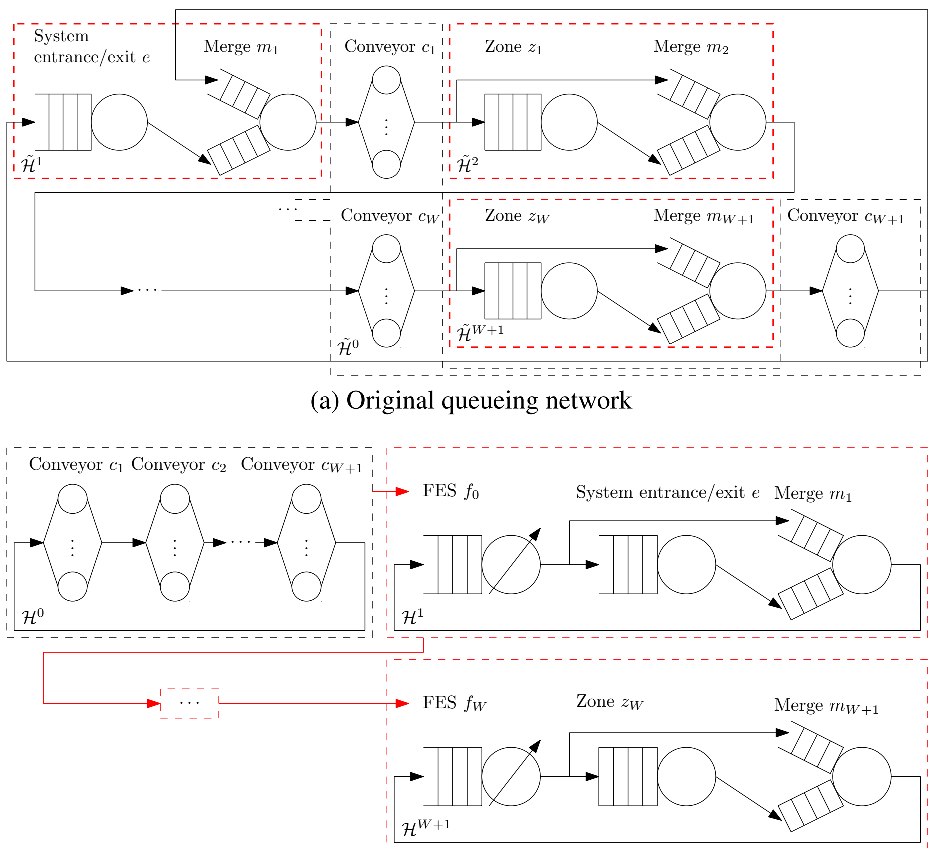
5.1 Aggregation technique

The aggregation technique was introduced by Chandy et al. [5] to study the performance of BCMP queueing networks [36]. The technique has been extended for more general multi-class queueing networks by Kritzinger et al. [45], Walrand [46], Hsiao and Lazar [47], and Boucherie and van Dijk [48]. Based on Norton's theorem, the idea of the aggregation technique is to decompose the queueing network into subnetworks and to replace each subnetwork by a flow equivalent single server with load-dependent service rates. The rates of the flow equivalent server (FES) are obtained by studying the subnetwork in isolation, i.e., by short-circuiting all nodes that are not in the subnetwork. The service rate of FES f_k when n totes are present is then taken equal to the throughput of the closed subnetwork with population n;

$$\mu_{f_k}(n) = X^k(n), \quad n = 1, \ldots, N. \tag{12}$$

The aggregation method is proven to be exact in case the queueing network has a product-form stationary distribution (see Chandy et al. [5]) and it can be used as a basis to analyze non-product form queueing networks [40].

In Figure 3a the queueing network of Section 4 is partitioned into $W+2$ subnetworks, $\{\tilde{\mathcal{H}}^0, \tilde{\mathcal{H}}^1, \ldots, \tilde{\mathcal{H}}^k, \ldots, \tilde{\mathcal{H}}^{W+1}\}$. The first subnetwork $\tilde{\mathcal{H}}^0$ contains all the conveyor nodes, whereas subnetwork $\tilde{\mathcal{H}}^k$, $k = 1, \ldots, W+1$ consists the entrance station or a zone, and a merge. The queueing network is analyzed by the approximate aggregation method as



(a) Original queueing network

(b) Approximation steps

Figure 3: The approximate aggregation technique applied on a zone picking system with W zones.

shown in Figure 3b, where the nodes are partitioned as follows;

$$\mathcal{H}^0 = \mathcal{C}, \tag{13}$$

$$\mathcal{H}^1 = \{f_0\} \cup \{e\} \cup \{m_1\}, \tag{14}$$

$$\mathcal{H}^{k+1} = \{f_k\} \cup \{z_k\} \cup \{m_{k+1}\}, \quad k = 1, \dots W. \tag{15}$$

The first step of the method is to study subnetwork $\mathcal{H}^0 = \tilde{\mathcal{H}}^0$ in isolation. Then for each subsequent subnetwork $\mathcal{H}^k$, the previous subnetwork is aggregated into FES f_{k-1} with service rates given by (12) and analyzed in isolation together with the nodes in $\tilde{\mathcal{H}}^k$. This process is repeated until the last subnetwork $\mathcal{H}^{W+1}$ from which the overall performance statistics are obtained. The performance of the individual nodes can now be calculated by disaggregating the network using the marginal queue length probabilities obtained from each subnetwork (see Section 5.2).

Our approximation method differs from other aggregation heuristics, e.g., Marie [49] and Neuse and Chandy [50]. These heuristics start by replacing each node that does not satisfy the product-form assumption by an equivalent node that does satisfy the assumption. By solving the subnetwork iteratively better estimates for the equivalent node are found until it resembles the original node up to a certain threshold. However, convergence might by slow and require many iterations, while our approach only requires a fixed amount of iterations equal to the number of subnetworks.

5.2 Algorithm

In this section, we summarize the approximation procedure for analyzing queueing network of Section 4. For a detailed description about the blocking probabilities in the jump-over network approximation the reader is referred to van der Gaast et al. [7]. The approximation procedure can now be summarized as follows:

Step 0: Initialize the blocking probabilities b_i, $i \in \mathcal{Z}$ to 0.

Step 1: Analyze the first subnetwork $\mathcal{H}^0$ for different population size $n = 1, 2, \dots, N$. For each n obtain the marginal queue length probabilities $\pi_i^0(m|n)$, $i \in \mathcal{H}^0$ that there are m totes in node i when the population size is n. Based on these results calculate the throughput $X^0(n)$, $n = 1, 2, \dots, N$.

Step 2: Construct FES f_k with the service rate given by $X^k(n)$, $n = 1, 2, \dots, N$. Now, analyze $\mathcal{H}^k$ for different population size $n = 1, 2, \dots, N$ using matrix-geometric methods and obtain the marginal queue length probabilities $\pi_i^k(m|n)$. from which calculate the throughput $X^k(n)$

Step 3: Go to Step 2 if $k < W + 1$. Otherwise estimate the new blocking probabilities and continue with Step 4.

Step 4: Go back to Step 1 in case there is no convergence of the blocking probabilities yet. Otherwise, calculate the performance statistics. In particular, the throughput rate of the system is given by $X(N) = X^{W+1}(N)$.

6 Numerical results

In this section we compare the results of our approximation method with a discrete-event simulation of the real queueing network. In Section 6.1 we test the performance of the approximation method for a zone picking system without recirculation. For each setup, the simulation model was run 10 times for 1,000,000 seconds, preceded by 10,000 seconds of initialization for the system to become stable, which guaranteed that the 95% confidence interval width of the average throughput is less than 1% of the mean value for all the runs. The maximum run time of the approximation was less than 5 minutes on a Core i7 with 2.4GHz and 8 GB of RAM.

6.1 Zone picking system with recirculation

In order to study the performance and accuracy of the algorithm of Section 5.2, we consider a zone picking system with 2, 4, or 6 zones. For a system with W zones, there are in total 2^W possible combinations of zones a tote can visit. We assume that each combination of zones (a class) has the same probability of being released into the system except for the empty set, e.g., $\psi_{\{\emptyset\}} = 0$ and $\psi_{\boldsymbol{r}} = 1/\left(2^W - 1\right)$. The time required to prepare a new tote to be launched into the system at the entrance station is equal to $\mu_e^{-1} = 5$ seconds. Each conveyor node requires a fixed deterministic conveyor times of $\mu_i^{-1} = 60$ seconds, $i \in \mathcal{C}$, whereas the time required to pass a merge node is equal to $\mu_i^{-1} = 3$ seconds, $i \in \mathcal{M}$. The time to pick products for a tote at a zone is $\mu_i^{-1} = 30$ seconds, $i \in \mathcal{Z}$. The number of order pickers in each zone d_i, $i \in \mathcal{Z}$ is equal to 1 and the input buffer sizes of each zone is respectively $q_i = 3$, $i \in \mathcal{Z}$. Finally, it is assumed that there is no output buffer after a zone and the entrance $(l_i = 0,\ i \in \mathcal{M})$. This means that only when the current tote has crossed the merge the order picker/entrance station can start to work on the next tote in line.

In Table 1 the results of the three configurations for both the approximation method and the simulation are shown in terms of the average throughput $X(N)$ (per hour^{-1}). We also show the results of the approximation method where the merge node is replaced by a single-server queueing node with an infinite buffer and the same service distribution as the current merge node (*no merging*). This means totes entering the merge are served at a first-come first-served basis and the order picker/entrance station is never blocked because of a full output buffer. This results in a simpler but potentially less accurate model to evaluate.

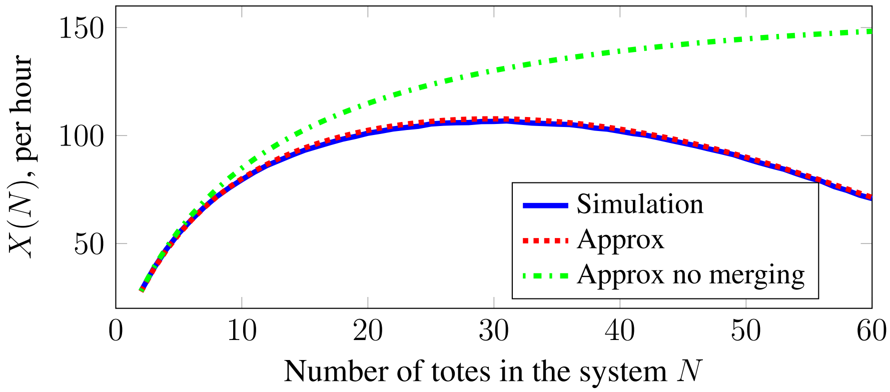
It can be seen that the approximation slightly overestimates the average throughput when reaching the maximum average throughput capability of the system. For example, for the configuration with two zones the maximum throughput capability that can reached

Table 1: Results of the average throughput $X(N)$ per hour of the approximation model and simulation for a zone picking system with 2, 4 and 6 zones with recirculation.

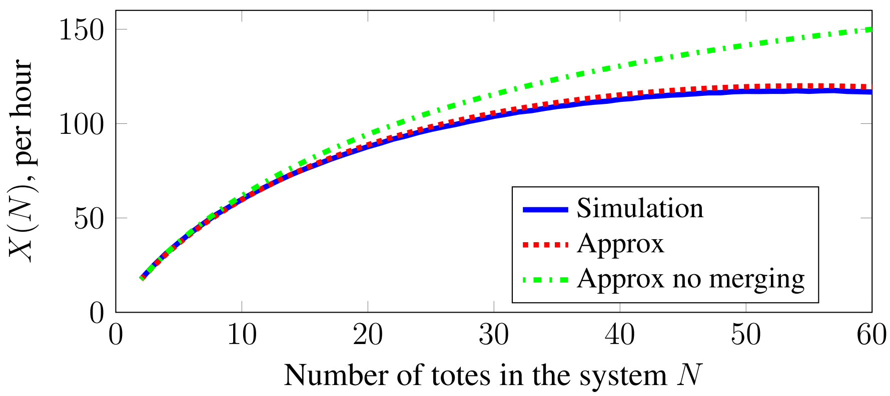
	2 zones			4 zones			6 zones		
N	Approx	Simulation	Error	Approx	Simulation	Error	Approx	Simulation	Error
5	54.33	54.73(±0.15)	-0.74	36.60	36.99(±0.11)	-1.05	26.68	27.17(±0.10)	-1.80
10	80.04	79.56(±0.21)	0.60	59.67	59.85(±0.17)	-0.30	44.76	45.34(±0.11)	-1.29
15	94.31	93.26(±0.22)	1.13	76.28	75.95(±0.29)	0.45	58.81	58.95(±0.13)	-0.24
20	102.42	101.08(±0.30)	1.33	88.77	87.79(±0.24)	1.12	70.20	69.98(±0.12)	0.32
25	106.55	105.37(±0.34)	1.12	98.33	96.91(±0.20)	1.47	79.64	79.07(±0.17)	0.71
30	107.74	106.53(±0.32)	1.13	105.65	103.89(±0.23)	1.70	87.54	86.54(±0.15)	1.16
35	106.46	105.28(±0.30)	1.12	111.19	109.07(±0.17)	1.94	94.20	92.89(±0.21)	1.41
40	102.96	102.00(±0.25)	0.95	115.23	112.84(±0.18)	2.12	99.80	98.28(±0.26)	1.54
45	97.40	96.54(±0.26)	0.89	117.98	115.40(±0.22)	2.23	104.50	102.70(±0.18)	1.75
50	89.98	89.28(±0.29)	0.79	119.54	117.10(±0.32)	2.09	108.39	106.32(±0.28)	1.95
55	81.10	80.45(±0.34)	0.80	120.01	117.12(±0.32)	2.47	111.57	109.29(±0.22)	2.09
60	71.36	70.84(±0.14)	0.73	119.45	116.75(±0.26)	2.31	114.09	111.66(±0.17)	2.18

is about 106 totes per hour when $N = 30$. For larges values, the average throughput starts to decrease due to the fact that totes flowing out of a zone have to wait a long time until they can be merged on the main conveyor, while at the same stopping the order picking from continuing his or her work on the next tote in line and the entrance station from releasing new totes. Totes on the main conveyor, on the other hand, recirculate until there is an open position in the input buffer of the zone. A similar effect can be seen in case of 4 and 6 zones.

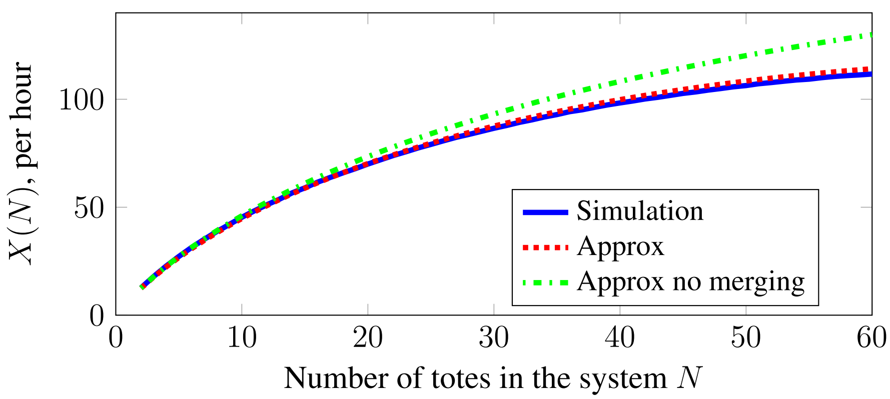
In Figure 4 the same results for the average throughput are shown again. In addition, the approximation is shown where the merges are replaced by first-come first-served single-server nodes. The figure shows that for the approximation without the merges large errors are made when the number of totes N becomes large. In fact, the throughput will never decrease since any additional tote that enters the system can always enter the conveyor. Eventually, the throughput stabilizes at a point that the utilization of the order pickers equals 1. It can be concluded that modeling the merge operation in detail is of great importance because otherwise the maximum throughput capability of the system cannot be determined correctly. This can lead to the expectation that the system has a much higher throughput capability than what in reality is possible.



(a) 2 zones



(b) 4 zones



(c) 6 zones

Figure 4: Results of the average throughput $X(N)$ per hour of the approximation model with and without merges modeled and simulation for 2, 4 and 6 zones without recirculation.

7 Conclusion

In this paper, we developed an analytical model for studying the merge operation in zone picking systems. We developed a queueing model that provides a valuable tool for rapid performance evaluations and design of complex zone picking systems in order to meet specific performance levels. It can be used to study and reduce delay caused by merges. Comparison of the approximation results to simulation for a wide range of parameters showed that the mean relative error for statistics as the system throughput is small.

A relevant extension is the situation where order pickers can help each other when the workload in one zone is high or leave when there is little work such that one order picker becomes responsible for picking products at multiple zones. Furthermore, the model may provide a starting point in order to approximate higher moments or the distribution of performance statistics such as the zone, segment, and, system throughput time.

References

[1] C. Petersen, Considerations in order picking zone configuration, International Journal of Operations & Production Management 22 (7) (2002) 793–805.

[2] J. Gu, M. Goetschalckx, L. McGinnis, Research on warehouse design and performance evaluation: A comprehensive review, European Journal of Operational Research 203 (3) (2010) 539–549.

[3] B. Park, Warehousing in the Global Supply Chain: Advanced Models, Tools and Applications for Storage Systems, Springer-Verlag, New York, 1 edn., 2012.

[4] C. Osorio, M. Bierlaire, An analytic finite capacity queueing network model capturing the propagation of congestion and blocking, European Journal of Operational Research 196 (3) (2009) 996–1007, ISSN 0377-2217.

[5] K. Chandy, U. Herzog, L. Woo, Parametric analysis of queuing networks, IBM Journal of Research and Development 19 (1) (1975) 36–42.

[6] G. Latouche, V. Ramaswami, Introduction to matrix analytic methods in stochastic modeling, vol. 5, Siam, 1999.

[7] J. van der Gaast, R. de Koster, I. Adan, J. Resing, Modeling and performance analysis of sequential zone picking systems, working Paper, 2012.

[8] R. De Koster, T. Le-Duc, K. Roodbergen, Design and control of warehouse order picking: A literature review, European Journal of Operational Research 182 (2) (2007) 481–501, ISSN 0377-2217.

[9] C. Petersen, An evaluation of order picking policies for mail order companies, Production and Operations Management 9 (4) (2000) 319–335.

[10] D. D. Yao, J. Buzacott, Modelling the performance of flexible manufacturing systems, International Journal of Production Research 23 (5) (1985) 945–959.

[11] A. E. Gray, U. S. Karmarkar, A. Seidmann, Design and operation of an order-consolidation warehouse: Models and application, European Journal of Operational Research 58 (1) (1992) 14–36.

[12] R. De Koster, Performance approximation of pick-to-belt orderpicking systems, European Journal of Operational Research 72 (3) (1994) 558–573, ISSN 0377-2217.

[13] C. J. Malmborg, Storage assignment policy tradeoffs, International Journal of Production Research 34 (2) (1996) 363–378.

[14] C. C. Jane, Storage location assignment in a distribution center, International Journal of Physical Distribution & Logistics Management 30 (1) (2000) 55–71.

[15] E. Jewkes, C. Lee, R. Vickson, Product location, allocation and server home base location for an order picking line with multiple servers, Computers & Operations Research 31 (4) (2004) 623–636, ISSN 0305-0548.

[16] M. Yu, R. De Koster, Performance approximation and design of pick-and-pass order picking systems, IIE Transactions 40 (11) (2008) 1054–1069.

[17] D. D. Eisenstein, Analysis and optimal design of discrete order picking technologies along a line, Naval Research Logistics (NRL) 55 (4) (2008) 350–362.

[18] J. C.-H. Pan, M.-H. Wu, A study of storage assignment problem for an order picking line in a pick-and-pass warehousing system, Computers & Industrial Engineering 57 (1) (2009) 261–268.

[19] M. Melacini, S. Perotti, A. Tumino, Development of a framework for pick-and-pass order picking system design, The International Journal of Advanced Manufacturing Technology 53 (9) (2010) 841–854, ISSN 0268-3768.

[20] W. Whitt, Approximating a point process by a renewal process, I: Two basic methods, Operations Research 30 (1) (1982) 125–147, ISSN 0030-364X.

[21] T. Kwo, A theory of conveyors, Management science 5 (1) (1958) 51–71.

[22] E. J. Muth, A model of a closed-loop conveyor with random material flow, AIIE Transactions 9 (4) (1977) 345–351.

[23] A. S. Bastani, E. Elsayed, Blocking in closed-loop conveyor systems connected in series with discrete and deterministic material flow, Computers & industrial engineering 11 (1) (1986) 40–45.

[24] A. S. Bastani, Analytical solution of closed-loop conveyor systems with discrete and deterministic material flow, European journal of operational research 35 (2) (1988) 187–192.

[25] R. L. Disney, Some multichannel queueing problems with ordered entry, Journal of Industrial Engineering 13 (1) (1962) 46–48.

[26] E. J. Muth, J. A. White, Conveyor theory: a survey, AIIE Transactions 11 (4) (1979) 270–277.

[27] D. Sonderman, An analytical model for recirculating conveyors with stochastic inputs and outputs, The International Journal Of Production Research 20 (5) (1982) 591–605.

[28] D. Sonderman, B. Pourbabai, Single server stochastic recirculation systems, Computers & operations research 14 (1) (1987) 75–84.

[29] E. Coffman Jr, E. Gelenbe, E. Gilbert, Analysis of a conveyor queue in a flexible manufacturing system, European journal of operational research 35 (3) (1988) 382–392.

[30] L. Schmidt, J. Jackman, Modeling recirculating conveyors with blocking, European Journal of Operational Research 124 (2) (2000) 422–436, ISSN 0377-2217.

[31] W. Zijm, I. Adan, R. Buitenhek, G. van Houtum, Capacity analysis of an automated kit transportation system, Annals of Operations Research 93 (1-4) (2000) 423–446.

[32] Y. Bozer, Y. Hsieh, Throughput performance analysis and machine layout for discrete-space closed-loop conveyors, IIE Transactions 37 (1) (2005) 77–89, ISSN 0740-817X.

[33] Y.-J. Hsieh, Y. Bozer, Analytical Modeling of Closed-Loop Conveyors with Load Recirculation, in: Computational Science and Its Applications - ICCSA 2005, vol. 3483 of *Lecture Notes in Computer Science*, Springer Berlin, 838–838, 2005.

[34] J. Jackson, Jobshop-like queueing systems, Management science 10 (1) (1963) 131–142.

[35] W. Gordon, G. Newell, Closed queuing systems with exponential servers, Operations Research 15 (2) (1967) 254–265, ISSN 0030-364X.

[36] F. Baskett, K. Chandy, R. Muntz, F. Palacios, Open, closed, and mixed networks of queues with different classes of customers, Journal of the ACM 22 (2) (1975) 248–260, ISSN 0004-5411.

[37] J. Buzen, Computational algorithms for closed queueing networks with exponential servers, Communications of the ACM 16 (9) (1973) 527–531, ISSN 0001-0782.

[38] M. Reiser, S. Lavenberg, Mean-value analysis of closed multichain queuing networks, Journal of the ACM 27 (2) (1980) 313–322, ISSN 0004-5411.

[39] R. M. Bryant, A. E. Krzesinski, M. S. Lakshmi, K. M. Chandy, The MVA priority approximation, ACM Transactions on Computer Systems (TOCS) 2 (4) (1984) 335–359.

[40] G. Bolch, S. Greiner, H. de Meer, K. Trivedi, Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications, Wiley-Interscience, 2 edn., ISBN 0471565253, 2006.

[41] N. Van Dijk, On Jackson's product form with "jump-over" blocking, Operations Research Letters 7 (5) (1988) 233–235, ISSN 0167-6377.

[42] B. Pittel, Closed exponential networks of queues with saturation: The Jackson-type stationary distribution and its asymptotic analysis, Mathematics of Operations Research 4 (4) (1979) 357–378, ISSN 0364-765X.

[43] R. Schassberger, Decomposable stochastic networks: Some observations, Modelling and Performance Evaluation Methodology 60 (1984) 135–150.

[44] A. Economou, D. Fakinos, Product form stationary distributions for queueing networks with blocking and rerouting, Queueing Systems 30 (3) (1998) 251–260, ISSN 0257-0130.

[45] P. S. Kritzinger, S. Van Wyk, A. E. Krzesinski, A generalisation of Norton's theorem for multiclass queueing networks, Performance Evaluation 2 (2) (1982) 98–107.

[46] J. Walrand, A Note on Norton's Theorem for Queuing Networks, Journal of Applied Probability 20 (2) (1983) 442–444.

[47] M.-T. T. Hsiao, A. A. Lazar, An extension to Norton's equivalent, Queueing Systems 5 (4) (1989) 401–411.

[48] R. J. Boucherie, N. M. van Dijk, A generalization of Norton's theorem for queueing networks, Queueing Systems 13 (1-3) (1993) 251–289.

[49] R. A. Marie, An approximate analytical method for general queueing networks, Software Engineering, IEEE Transactions on 5 (5) (1979) 530–538.

[50] D. Neuse, K. M. Chandy, HAM: The heuristic aggregation method for solving general closed queueing network models of computer systems, in: ACM SIGMETRICS Performance Evaluation Review, vol. 11, ACM, 195–212, 1982.