Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Conveyor Merges in Zone Picking Systems: A Tractable and Accurate Approximate Model

J. P. van der Gaast, M. B. M. de Koster, I. J. B. F. Adan

informs
http://pubsonline.informs.org/journal/trsc/

TRANSPORTATION SCIENCE
Vol. 52, No. 6, November–December 2018, pp. 1428–1443
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

Conveyor Merges in Zone Picking Systems: A Tractable and Accurate Approximate Model

J. P. van der Gaast,[a] M. B. M. de Koster,[b] I. J. B. F. Adan[c]

[a] Rijksuniversiteit Groningen, 9747 AE Groningen, Netherlands; [b] Department of Technology and Operations Management, Erasmus Universiteit, 3062 PA Rotterdam, Netherlands; [c] Department of Industrial Engineering and Innovation Sciences, Technische Universiteit Eindhoven, 5612 AZ Eindhoven, Netherlands

Contact: j.p.van.der.gaast@rug.nl, http://orcid.org/0000-0001-7591-1552 (JPvdG); rkoster@rsm.nl, http://orcid.org/0000-0002-1740-7822 (MBMdK); i.j.b.f.adan@tue.nl (IJBFA)

Received: March 6, 2015
Revised: October 7, 2016
Accepted: May 2, 2017
Published Online in Articles in Advance: October 10, 2018

https://doi.org/10.1287/trsc.2017.0782

Abstract. Sequential zone picking systems are popular conveyor-based picker-to-parts order picking systems that divide the order picking area in work zones. When designing a zone picking system, it is important to know whether the throughput capability of the system can meet customer demand. However, the performance and maximum throughput capability of a zone picking system is largely determined by congestion and blocking that occur at the various conveyor merges in the system. In this paper we develop an analytical model to study the impact of conveyor merges in sequential zone picking systems. Because of finite buffers, blocking, recirculation, and merging, the resulting queueing model does not have a product-form stationary queue-length distribution which makes exact analysis practically infeasible. Therefore, we develop an approximate solution by using an aggregation technique and matrix-geometric methods to study the throughput capability of the system. The model is suitable to support rapid design of complex zone picking systems, in terms of number and length of zones, input and output buffer capacities, and storage allocation of products to zones to meet prespecified performance targets. Comparison of the approximation results to simulation show that for a wide range of parameters the mean relative error in the system throughput is typically less than 5%. The model accurately predicts the loss in throughput due to congestion and blocking at the merges, and can be used to allocate input and output buffer spaces to maximize the throughput capability of the system.

Keywords: warehousing • queueing theory • material handling • logistics

1. Introduction

Conveyor systems are a critical component of many order picking and sorting systems responsible for moving products from one location to another. One of the most important functions of conveyor systems is to consolidate multiple flows of products into one single flow (a *merge* operation). These merges are often potential points of congestion that can lead to blocking and increased order throughput times. Obviously, the performance of the merges strongly influences the overall system performance. In *sequential zone picking*, a very popular order picking method in practice, conveyor merges frequently occur and must be considered when determining the maximum throughput capability of the system.

When a zone picking system is under heavy load, congestion and blocking can occur at the conveyor merges due to limited free space on the conveyor. This congestion leads to reduced throughput and causes unpredictable throughput times. Previous studies have reported that the throughput can drop considerably (see Dallery and Gershwin 1992 for an overview on production lines). As a direct consequence, orders cannot be shipped on time, which leads to delayed customer deliveries and revenue loss. However, estimating the throughput of a zone picking system, or any conveyor system with one or multiple merges, is very complicated due to congestion at the merges and proliferation of congestion over the merges. This holds even stronger in the presence of job variability, variability of the performance of components, and human operators.

This paper attempts to quantify the impact of merge operations on the throughput of zone picking systems to determine their maximum throughput capability. The system is therefore modeled as a closed queueing network that describes the conveyor, the pick zones, and the merge locations. Because of finite buffers, blocking, recirculation, and merging, the resulting queueing model does not have a product-form stationary queue-length distribution; this makes exact analysis practically infeasible. Therefore, we develop a new performance approximation model to study conveyor merges using an aggregation technique (Chandy, Herzog, and Woo 1975) and matrix-geometric methods (Latouche and Ramaswami 1999). We show that the approximation model, which includes priority nodes and blocking, produces very accurate estimates of the maximum throughput capability of a 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 input and output buffer lengths, and maximum number of totes in the systems. Our results show that throughput drops dramatically when congestion and blocking at the merges increase, and that if the number of totes in the system increases, it becomes more beneficial to increase the size of the output buffer rather than the input buffer of the zones. Finally, our methods to study conveyor merges in zone picking systems can also be applied to other internal logistic systems such as automated baggage handling, sorting systems, vehicle-based automated guided vehicle (AGV) systems, and production lines with conveyors.

The organization of this paper is as follows. In Section 2, we discuss zone picking systems. An overview of existing models for 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 we explain our approximation method; we verify its performance in Section 6 via computational experiments for a range of parameters. In Section 7, we conclude and suggest some extensions of the model.

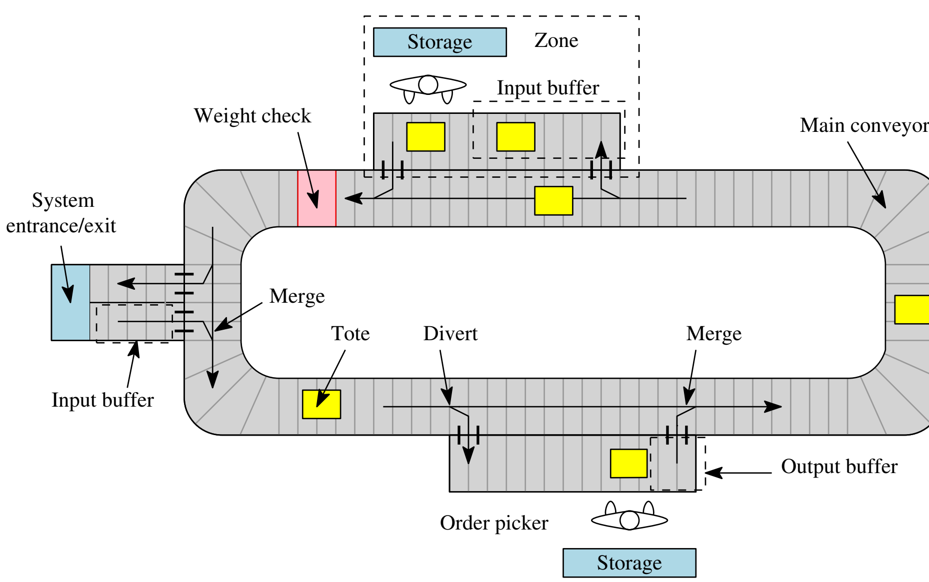
2. Zone Picking Systems

Zone picking is a picker-to-parts order picking method that divides the order picking area in work zones, each operated by one or multiple order pickers, each responsible for picking products only from her zone (Petersen 2002; Gu, Goetschalckx, and McGinnis 2010). The major advantages of zone picking systems are the high-throughput ability, scalability, and flexibility in handling various order volumes and product sizes, with a varying number of order pickers (Van der Gaast et al. 2012). These systems are often applied in e-commerce warehouses handling customer orders with a large number of order lines and with many different products kept in stock (Park 2012).

Zone picking systems can be categorized in *parallel* or *sequential* zone picking (De Koster, Le-Duc, and Roodbergen 2007). In a parallel zone picking system, multiple pickers in multiple zones can simultaneously work 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. In sequential zone picking (also called pick-and-pass systems), an order is assigned to an order tote or order carton that travels on the conveyor and is sequentially passed to the next zone where order lines that must be added to the order may (or may not) be stored. The advantage of sequential zone picking is that order integrity is maintained and no sorting and product consolidation is required (Petersen 2000). These advantages make sequential zone picking systems highly popular in practice, especially in e-commerce warehouses. In this paper, we only consider sequential zone picking (hereafter zone picking).

Figure 1 is a schematic representation of a zone picking system, where the picking area is divided into two zones. These zones are connected by conveyors enabling automatic transportation of customer orders through the system. A customer order is released into the system at the system entrance as an order tote, which contains a list of products to be picked and their locations within the picking area. After release, the tote travels on the main conveyor to the required zones, i.e., where a product that must still be picked for the tote's order is stored. When an order reaches a zone

Figure 1. (Color online) A Zone Picking System with Two Zones



it must visit, it is automatically diverted from the conveyor into the input buffer of the zone. Eventually, the order picker starts picking the required products that are stored 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 so that it can travel to the next zone. Orders that do not require products from the zone remain on the conveyor and are transported to the next zone in line. When the tote has visited all of the required zones, it leaves the system at the exit and, if available, a new order tote is immediately released into the system.

There are many different configurations of zone picking systems. These variants include, e.g., workstation type, pick-face design, buffer lengths, storage system layout, and conveyor configuration. The conveyor configuration is of especially great importance, since it affects how and when totes arrive at the zones. In most zone picking systems, a tote can skip a zone if it does not need to pick up any order lines from the zone. Also, combined with a closed-loop conveyor, totes can skip a zone if the zone's input buffer is fully occupied. The tote can then return to this zone after visiting other zones or after recirculating on the conveyor (a weight check at the end of the conveyor ensures that the tote does not leave the system before visiting all of 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, see Van der Gaast et al. (2012).

The *throughput* is the key performance indicator in zone picking systems. Throughput, measured as the number of completed orders or order lines per period of time, is used to judge whether the order picking system is capable of meeting a certain customer demand. It is also used to determine the cut-off time so that orders are guaranteed to be shipped during the next delivery cycle. E-commerce warehouse companies must especially address very strict delivery lead times since the customers demand fast delivery, often within 24 hours.

The maximum throughput capability of a zone picking system is largely determined by the performance of the merges. After entering the system and after each zone, totes merge on the main conveyor to move to their next location. At a merge location, totes that are already on the main conveyor have absolute priority (the main conveyor moves continuously without possibilities for accumulation). Therefore, to allow a tote to leave the output buffer, its predecessors from the same buffer should have left, and a sufficiently large space on the main conveyor must be available to prevent collisions. Under low use, the time required for a sufficiently large space on the conveyor to show up is negligible. However, many systems are highly used during peak hours, e.g., in e-commerce environments. In such a case, this space can become scarce leading to long merge times and a loss in overall system performance. In addition, the output buffer can become full and cause the order picker or the entrance station to stop work on the next tote in line. The order picker or entrance station can resume its work only if there is at least one empty place in the output buffer. Finally, some zone picking systems do not have an output buffer; in these cases, the order picker or entrance station must always wait until the processed tote has entered the main conveyor before starting to work on the next tote in line.

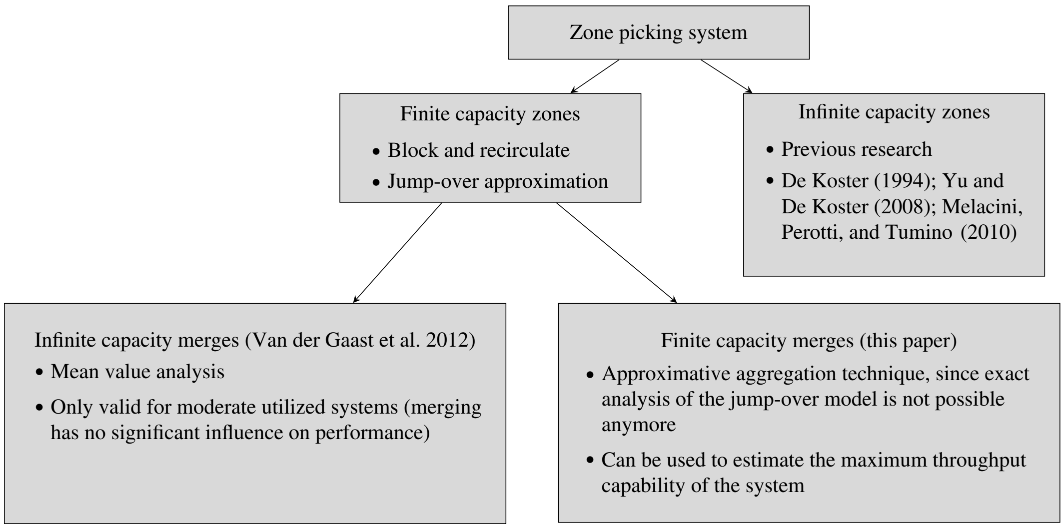
3. Literature Review

Zone picking systems can be analyzed by simulation models and by testing various scenarios. Although simulation allows for very accurate modeling, it is time consuming to build and evaluate each scenario or layout design, especially when the system is highly used and blocking frequently occurs. Also, the accuracy of the simulation strongly depends on the quality of the calibration data (Osorio and Bierlaire 2009). Petersen (2002) 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 average walking distances.

Another approach to analyze zone picking systems are stochastic models. Stochastic models facilitate study of the random fluctuations that occur in zone picking systems, e.g., at peak periods during the day since at those times more customer orders are being received and the cut-off delivery time is approaching, or because of the human factors in the picking process. Stochastic models are, in general, much faster, more flexible, and less data expensive in estimating the performance of a zone picking system. They can be used as evaluation tools in the initial design phase to help designers quickly evaluate many design alternatives and to narrow the available design space (Gu, Goetschalckx, and McGinnis 2010). They can also be used to optimize the system in later design (or control) phases in terms of order release rules and workload allocation.

The analysis of conveyor systems has received much attention. Initially deterministic conveyor models were developed to study feasibility conditions, such as loading/unloading rates and conveyor lengths (Kwo 1958; Muth 1977; Bastani and Elsayed 1986; Bastani 1988). However, these models fail to capture the effects that random fluctuations in the input and/or output can have on the design and performance of a conveyor system. Therefore, stochastic models for conveyor systems were introduced by, e.g., Disney (1962), and a

Figure 2. Differences Between the Current Paper and Previous Research



conveyor system as a multichannel queueing system with ordered entry (Sonderman 1982). Afterwards, stochastic models were applied to analyze a wide variety of different material handling systems (see, e.g., Coffman Jr, Gelenbe, and Gilbert 1988; Schmidt and Jackman 2000; Zijm et al. 2000; Bozer and Hsieh 2005; Hsieh and Bozer 2005).

Although literature on zone picking systems is still very limited, it has gained popularity in the past two decades. De Koster (1994) 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 (1996) and Jane (2000) developed models for product allocation to study the trade-offs in space requirements and retrieval costs with dedicated and randomized storage in a zone picking system and workload balancing between zones. Jewkes, Lee, and Vickson (2004) and Eisenstein (2008) extended this by considering the location of the picker home base to minimize the expected order cycle time. Yu and De Koster (2008) analyzed zone picking systems without recirculation and presented an approximation method based on a $G/G/m$ queueing network. Melacini, Perotti, and Tumino (2010) modeled a zone picking system as a network of queues. To estimate performance statistics, such as the use, throughput rate of a zone, and the mean and standard deviation of the throughput time of the totes, they used Whitt's queueing network analyzer (Whitt 1982). Van der Gaast et al. (2012) studied a single/multisegment zone picking system with the block-and-recirculation protocol. They showed that the system can be very accurately approximated by a related product-form queueing network with the jump-over protocol. A limitation of this work is that the conveyor merges are not modeled, which only facilitates the study of the moderate loaded zone picking system. This is because if the number of totes in this model is increased, the system throughput would stabilize as the main conveyor would act as a buffer for the additional totes in the system. Here we use this framework to analyze zone picking systems with a block-and-recirculate protocol and extend it by a new approximation analysis that facilitates the study of finite capacity conveyor merges and consequently determines the maximum throughput capability of the system.

To summarize, in Figure 2 we explain the differences between our current paper and the previous research. To our knowledge, previous research does not facilitate determination of the maximum throughput capability of a zone picking system. In Section 4, we present a model that can be used for this analysis.

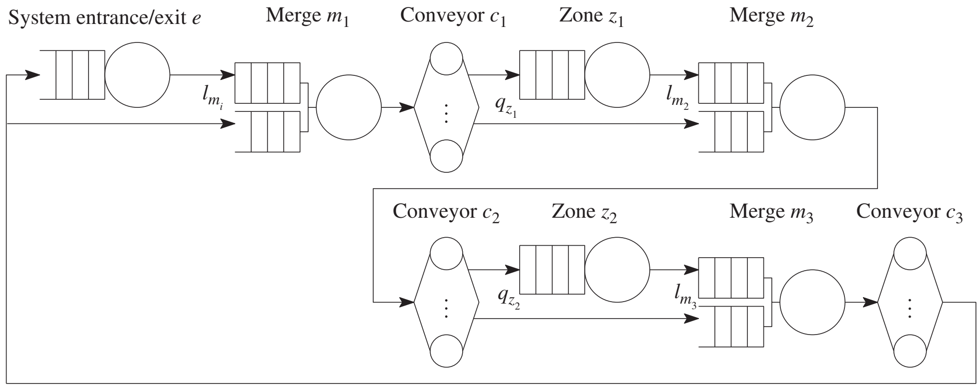
4. Queueing Model for Zone Picking Systems

Figure 3 shows the model for zone picking systems with merges for the case of two zones. Van der Gaast et al. (2012) studied a similar model, but they did not 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 as follows: The system entrance/exit is denoted as e; $\mathscr{Z} = \{z_1, \ldots, z_W\}$ denotes the set of zones; $\mathscr{M} = \{m_1, \ldots, m_{W+1}\}$ denotes the set of merges; and $\mathscr{C} = \{c_1, \ldots, c_{W+1}\}$ is the set of conveyors in the network. Finally, set $\mathscr{S} = \{e\} \cup \mathscr{C} \cup \mathscr{M} \cup \mathscr{Z}$ is the union of all of 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

Figure 3. The Corresponding Queueing Network with System Entrance/Exit Station e, Conveyors $\mathscr{C} = \{c_1, c_2, c_3\}$, Merges $\mathscr{M} = \{m_1, m_2, m_3\}$, and Zones $\mathscr{Z} = \{z_1, z_2\}$



be immediately replaced by a new tote. Each tote has a class $\mathbf{r} \subseteq \mathscr{Z}$, e.g., $\mathbf{r} = \{z_2, z_3\}$ 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 which is the rate at which a tote is prepared to enter the system (unfolding, adding labels and packing list, etc.). This is a valid assumption for zone picking system design, which aims to study the throughput capacity of the system. Operational issues, such as the effect of varying customer order arrival rates and customer order waiting times are not within the scope of this paper, although they could be studied using a similar approach. In addition, zone picking systems apply a workload control mechanism (Park 2012). This mechanism sets an upper bound on the number of totes in the system and only releases a new tote when a picked tote leaves the system.

• The conveyor nodes are assumed to be infinite-server nodes with a deterministic delay $1/\mu_i$, $i \in \mathscr{C}$.

• Each zone has one order picker. The order picking time is assumed to be exponentially distributed with rate μ_i, $i \in \mathscr{Z}$, that captures both variations in the pick time per order line and variations in the number of order lines to be picked. The assumption of the exponential distribution can be relaxed to a phase-type distribution at the cost of a more complex state space. The same holds for the number of order pickers per zone.

• When the order picker is busy, incoming totes at her station are stored in a finite input buffer of size $q_i(\geq 0)$, $i \in \mathscr{Z}$. 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 nonidentical repeat 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, for low or high priority totes, is assumed to be exponentially distributed with rate μ_i^H for high priority totes and μ_i^L for low priority totes, $i \in \mathscr{M}$. Similar to the zones, the assumption of the exponential distribution can be relaxed.

• Each merge has a limited capacity of size $l_i(\geq 0)$, $i \in \mathscr{M}$ 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 the low priority tote must wait at its current node, subsequently blocking the order picker/entrance station from starting to work on the next tote in line. We can distinguish two distinct cases for the unblocking procedure. In case $l_i \geq 1$, the tote leaves its current node and unblocks the order picker/entrance station only when there is at least one open position for low priority totes at the merge node. Whenever there is no output buffer ($l_i = 0$), the order picker/entrance station only unblocks after the current tote has passed the merge.

Let $\mathbb{S}(N)$ be the state space of the network with states $\mathbf{x} = (\mathbf{x}_i;\, i \in \mathscr{S})$. For each of these states, it holds that the number of totes in the system equals $\sum_{i\in\mathscr{S}} n_i = N$, where n_i, $i \in \mathscr{S}$ is the number of totes in node i. The state of node $i \in \mathscr{C}$ is $\mathbf{x}_i = (\mathbf{r}_{i1}, \ldots, \mathbf{r}_{in_i})$, with $\mathbf{r}_{i1}$ as the class of the first tote in the node, and $\mathbf{r}_{in_i}$ as the class of the last tote in the node. The state of node $i \in \{e\} \cup \mathscr{Z}$ is $\mathbf{x}_i = (\mathbf{r}_{i1}, \ldots, \mathbf{r}_{in_i}; y_i)$ where $y_i = 1$ if the order picker/entrance is blocked since a tote has finished service but cannot leave the zone because the merge is occupied and $y_i = 0$ otherwise. The state of merge node $i \in \mathscr{M}$, is defined as $\mathbf{x}_i = (\mathbf{r}_{i1}^H, \ldots, \mathbf{r}_{in_i^H}^H; \mathbf{r}_{i1}^L, \ldots, \mathbf{r}_{in_i^L}^L)$ with $n_i = n_i^H + n_i^L$ as the number of totes with high and low priority, respectively. The number of low priority totes in each merge should not exceed the capacity of the merge's output buffer l_i; $n_i^L \leq l_i$, $i \in \mathscr{M}$. Finally, the number of totes in each zone satisfies the capacity constraint $n_i \le q_i + 1$, $i \in \mathcal{Z}$, which implies that a tote cannot enter the zone if the input buffer is full and the order picker is occupied/blocked.

Denote by $p_{i\mathbf{r},j\mathbf{s}}(\mathbf{x})$ the state dependent routing probability that a tote of class $\mathbf{r}$ is routed from node i to node j and enters as a class $\mathbf{s}$ tote given that the network is in state $\mathbf{x}$. A new tote of class $\mathbf{r} \subseteq \mathcal{Z}$ is released at the system entrance with given probability $\psi_{\mathbf{r}}$. These release probabilities correspond to a known order profile that can be obtained using historical order data or forecasts. After release, a tote of class $\mathbf{r}$ moves from the system entrance to the first merge node m_1 with low priority. In general, after merging, a tote travels to the conveyor node c_i. After transport, at conveyor node c_i, the tote will enter the input buffer of zone z_i if $z_i \in \mathbf{r}$ and its input buffer is not full, or move to the next merge m_{i+1} with high priority. In case the tote needs to enter and the buffer is full, the tote skips the zone and moves to the next merge m_{i+1}, while it keeps the same class. If the buffer is not full, the tote enters the input buffer of the zone and, after possibly waiting in the input buffer in case the order picker still has to pick for other order totes first, the order picker picks the required order lines. After all picks are completed, the tote will enter the merge node with low priority and changes its class to $\mathbf{s} = \mathbf{r}\backslash\{z_i\}$. 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 of the totes with $\mathbf{r} \ne \varnothing$ are routed to the first merge node m_1 with high priority; the other totes with $\mathbf{r} = \varnothing$ move to the exit and are immediately replaced by a new tote that is waiting for release at the entrance.

Then, the routing probabilities of the network can be written as follows:

$$p_{e\varnothing,m_1\mathbf{r}}(\mathbf{x}) = \psi_{\mathbf{r}}, \tag{1}$$

$$p_{m_i\mathbf{r},c_i\mathbf{r}}(\mathbf{x}) = 1, \quad i = 1,\ldots,W, \tag{2}$$

$$p_{c_i\mathbf{r},z_i\mathbf{r}}(\mathbf{x}) = 1, \quad i = 1,\ldots,W,\ z_i \in \mathbf{r} \text{ and } n_{z_i} < q_{z_i} + 1, \tag{3}$$

$$p_{c_i\mathbf{r},m_{i+1}\mathbf{r}}(\mathbf{x}) = 1, \quad i = 1,\ldots,W,\ z_i \notin \mathbf{r} \text{ or } n_{z_i} = q_{z_i} + 1, \tag{4}$$

$$p_{z_i\mathbf{r},m_{i+1}\mathbf{s}}(\mathbf{x}) = 1, \quad i = 1,\ldots,W,\ \mathbf{s} = \mathbf{r}\backslash\{z_i\}, \tag{5}$$

$$p_{c_{W+1}\varnothing,e\varnothing}(\mathbf{x}) = 1, \tag{6}$$

$$p_{c_{W+1}\mathbf{r},m_1\mathbf{r}}(\mathbf{x}) = 1, \quad \mathbf{r} \ne \varnothing, \tag{7}$$

where every other probability is equal to 0.

Exact analytic methods to analyze queueing networks are only known for a limited set of models that satisfy certain conditions. The majority of these models have a *product-form* stationary distribution (Jackson 1963; Gordon and Newell 1967; Baskett et al. 1975). For an extensive review on product-form solutions in queueing networks, see the comprehensive books of Van Dijk (1993); Chao, Miyazawa, and Pinedo (1999); and Serfozo (1999). 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 (Buzen 1973) and the mean-value analysis (MVA) (Reiser and Lavenberg 1980) 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 (Bryant et al. 1984), and because of the dynamic block-and-recirculate protocol (Van der Gaast et al. 2012). Both elements cause the independence assumption to be violated. Also, direct analysis of the resulting underlying Markov chain is not feasible due to state-space explosion, which prevents analysis of the Markov chain within reasonable time and storage. Usually, nonproduct-form queueing networks are studied using approximation analysis. An overview of many general techniques is presented in Bolch et al. (2006).

Van der Gaast et al. (2012) show that the queueing network without merges can be 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 match the flows of the original network. This is done by introducing a Bernoulli process that randomly determines for every tote that intends to visit z_i, $i \in \mathcal{Z}$, independently of whether the tote actually visited z_i, or whether the tote should return to z_i. The probability b_i of the Bernoulli process that a tote should return to z_i 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, the blocking probabilities are not known in advance, but they are iteratively estimated after an initial guess from the approximation by calculating the probability that the zone is fully occupied in the approximation network.

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, routing probabilities (1)–(7) become state independent. This means after service at c_i, each tote with $z_i \in \mathbf{r}$ is routed to z_i regardless of whether the buffer of the zone is full (8)–(9). The tote will enter the buffer if it is not full; otherwise the tote skips the node. Then for each class $\mathbf{r}$ tote, independent of whether the tote visited or skipped z_i (because of a full buffer), $p_{z_i\mathbf{r},m_{i+1}\mathbf{r}} = b_{z_i}$ and $p_{z_i\mathbf{r},m_{i+1}\mathbf{s}} = 1 - b_{z_i}$, $i = 1,\ldots,M$, where $\mathbf{s} = \mathbf{r}\backslash\{z_i\}$. This means that a tote of class $\mathbf{r}$ is tagged as *skipped* z_i and routed to the next merge node m_{i+1} with the same class with probability b_{z_i}; otherwise, the tote is tagged as *visited* z_i with probability $1 - b_{z_i}$ and the class of the tote changes to $\mathbf{s} = \mathbf{r}\backslash\{z_i\}$. Summarizing, the routing probabilities (3)–(5) are replaced by

$$p_{c_i\mathbf{r},z_i\mathbf{r}} = 1, \quad i = 1,\ldots,W,\ z_i \in \mathbf{r}, \tag{8}$$
$$p_{c_i\mathbf{r},m_{i+1}\mathbf{r}} = 1, \quad i = 1,\ldots,W,\ z_i \notin \mathbf{r}, \tag{9}$$
$$p_{z_i\mathbf{r},m_{i+1}\mathbf{r}} = b_{z_i}, \quad i = 1,\ldots,W, \tag{10}$$
$$p_{z_i\mathbf{r},m_{i+1}\mathbf{s}} = 1 - b_{z_i}, \quad i = 1,\ldots,W,\ \mathbf{s} = \mathbf{r}\backslash\{z_i\}. \tag{11}$$

Because the recirculation process is made independent of the state of the buffer, the block-and-recirculate protocol is replaced by the *jump-over* blocking protocol (Van Dijk 1988). Under this protocol, each tote of class $\mathbf{r}$ leaving z_i, 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 (Pittel 1979; Schassberger 1984; Van Dijk 1988; Economou and Fakinos 1998).

However, this jump-over network still has no product-form due to the finite capacity priority queues. In this paper, we develop a new decomposition-based approximation by studying each merge and zone location in isolation. Our method progressively aggregates parts of the network and replaces the aggregated subnetwork by a flow equivalent single node. The approximation directly solves the global balance equations of the underlying Markov chain of the subnetworks for its steady-state distribution.

5. Approximate Aggregation Method

5.1. Aggregation Technique

The aggregation technique was introduced by Chandy, Herzog, and Woo (1975) to study the performance of BCMP queueing networks (Baskett et al. 1975). The technique has been extended for more general multiclass queueing networks by Kritzinger, Van Wyk, and Krzesinski (1982), Walrand (1983), Hsiao and Lazar (1989), and Boucherie and van Dijk (1993). Based on Norton's theorem, the idea of the aggregation technique is to decompose the queueing network into subnetworks and to replace each subnetwork with 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 the kth FES f_k when n totes are present is set equal to $X^k(n)$, the throughput of the closed subnetwork where the number of totes is n

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

The aggregation method is proven to be exact if the queueing network has a product-form stationary distribution (Chandy, Herzog, and Woo 1975), and can be used as a basis to analyze nonproduct-form queueing networks (Bolch et al. 2006).

Figure 4(a) presents the queueing network of Section 4. It is analyzed by the approximate aggregation method as shown in Figure 4(b), where the nodes are partitioned into $W+2$ subnetworks as follows:

$$\mathscr{H}^0 = \mathscr{C}, \tag{13}$$
$$\mathscr{H}^1 = \{f_0\} \cup \{e\} \cup \{m_1\}, \tag{14}$$
$$\mathscr{H}^{k+1} = \{f_k\} \cup \{z_k\} \cup \{m_{k+1}\}, \quad k = 1,\ldots W. \tag{15}$$

The first step of the approximation is to determine the chain visit ratios V_i that a tote visits node i

$$V_i = \frac{\sum_{\mathbf{r}\subseteq\mathscr{Z}} \lambda_{i\mathbf{r}}}{\sum_{\mathbf{r}\subseteq\mathscr{Z}} \lambda_{e\mathbf{r}}}, \quad i \in \mathscr{S}, \tag{16}$$

where $\lambda_{i\mathbf{r}}$ is the class dependent visit ratio of a class $\mathbf{r}$ tote to node i satisfying the traffic equations (up to a multiplicative constant)

$$\lambda_{i\mathbf{r}} = \sum_{j\in\mathscr{S}} \sum_{\mathbf{s}\subseteq\mathscr{Z}} \lambda_{j\mathbf{s}} p_{j\mathbf{s},i\mathbf{r}}, \quad i \in \mathscr{S}, \mathbf{r} \subseteq \mathscr{Z}. \tag{17}$$

The next step is to study subnetwork $\mathscr{H}^0$ in isolation. Because subnetwork $\mathscr{H}^0$ only consists of conveyor nodes with deterministic service, the average throughput of the subnetwork with number of totes n is simply given by

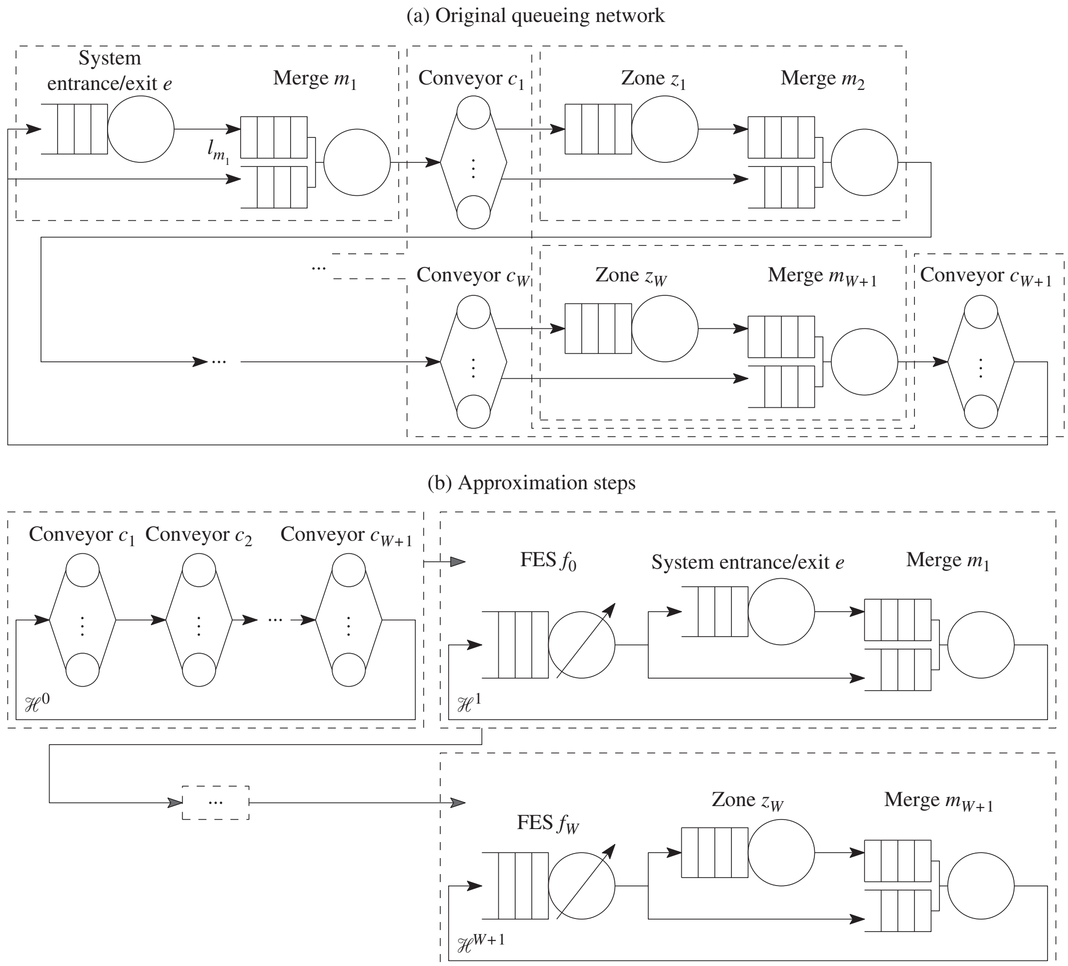
$$X^0(n) = \sum_{i\in\mathscr{H}^0} \frac{n}{V_i}\mu_i, \quad n = 1,\ldots,N. \tag{18}$$

The marginal queue-length probabilities $\pi_i^0(j \mid n)$, $n = 1,\ldots,N$, $i \in \mathscr{H}^0$, that there are j totes in node i in subnetwork $\mathscr{H}^0$ given that the number of totes is n can be similarly calculated as a service center of Type-3 (infinite server with general distributed service times) in a BCMP network (Baskett et al. 1975).

Then for each subsequent subnetwork $\mathscr{H}^k$, 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 $\mathscr{H}^k$. For each of these networks, we only need to know the visit ratio of totes that visit the entrance/zone or not, and the ratio of totes that skip the zone, which depends on b_{z_i}. This process is repeated until the last subnetwork $\mathscr{H}^{W+1}$, from which the overall performance statistics such as the throughput 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 5.3).

Our approximation method differs from other aggregation heuristics, e.g., Marie (1979) and Neuse and Chandy (1982). These heuristics start by replacing each

Figure 4. The Approximate Aggregation Technique Applied to a Zone Picking System with W Zones



node that does not satisfy the product-form assumption with an equivalent product-form node on an initial guess. This equivalent network can be easily analyzed using any of the efficient product-form algorithms such as MVA. To obtain a better approximation for each nonproduct-form node, a two-queue network is constructed, consisting of the original nonproduct-form node and a single aggregated node containing all of the other nodes in the equivalent network. The results of the analysis of this two-queue network are then used to make a better estimate for the nonproduct-form node in the equivalent network. This is iteratively repeated until all of the equivalent product-form nodes resemble the original nodes up to a certain prespecified threshold. However, convergence might be slow and many iterations may be required, while the number of iterations of our approach is equal to the number of subnetworks, and the underlying Markov chain is solved only once.

Section 5.2 shows how the other subnetworks $\mathscr{H}^k$, $k = 1, \ldots, W+1$ can be studied. The full approximate method is presented in Section 5.3.

5.2. Solving Subnetworks $k \geq 1$

In this section, we describe the analysis of the subnetwork $\mathscr{H}^1$ till $\mathscr{H}^{W+1}$. Each of these subnetworks consists of a node with preemptive nonidentical repeat priority, i.e., the merge, and thus cannot be analyzed using conventional product-form solution techniques (Bryant et al. 1984). Using aggregation, we reduce the size of the problem to a small system that we can efficiently model as a finite Markov process and directly solve the global balance equations of the underlying Markov chain for its steady-state. This allows us to calculate for a given $n = 1, \ldots, N$ the throughput $X^k(n)$ of subnetwork $\mathscr{H}^k$.

Let subnetwork $\mathscr{H}^k$, $k = 1, \ldots, W+1$ be described by a Markov process with state space $\mathbb{W}^k(n)$ with states (i, j, l) and the number of totes in the subnetwork is n. The state variable i denotes the number of totes waiting at the input buffer or in service in e or z_{k-1}, state variable j represents the number of totes with low priority at merge m_k, and includes the totes that finished service in e or z_{k-1} but cannot enter the output buffer or cross the merge. Finally, state variable l denotes the number of totes with high priority at the merge. Note that the number of totes at the FES f_{k-1} for any state is implicitly given by $u = n - i - j - l$.

Let the transition rates from state (i, j, l) to state (i', j', l') be given by $q_{(i,j,l)(i',j',l')}$. For a tote leaving FES f_k, the rates for $\mathcal{H}^k$, $k > 1$ can be written as follows:

$$q_{(i,j,l)(i+1,j,l)} = V_{z_{k-1}}\mu_{f_{k-1}}(u), \quad u < n,\ i + y_{z_{k-1}} < q_{z_{k-1}} + 1, \tag{19}$$

$$q_{(i,j,l)(i,j,l+1)} = [V_{m_k} - V_{z_{k-1}}]\mu_{f_{k-1}}(u), \quad u < n, i + y_{z_{k-1}} < q_{z_{k-1}} + 1, \tag{20}$$

$$q_{(i,j,l)(i,j,l+1)} = V_{m_k}\mu_{f_{k-1}}(u), \quad u < n,\ i + y_{z_{k-1}} = q_{z_{k-1}} + 1. \tag{21}$$

Transition rate (19) is the rate at which a tote from FES f_{k-1} enters the zone. A tote can only enter the input buffer if the number of totes at the zone is lower than the zone's maximum capacity $q_{z_{k-1}} + 1$. The number of totes currently in z_{k-1} equals i plus an additional tote $y_{z_{k-1}} = 1_{\{j=l_{m_k}+1\}}$, where the indicator function $1_{\{\cdot\}}$ equals one if there is a tote that just received service and is waiting to leave the zone, but cannot since the output buffer is full ($j = l_{m_k} + 1$). Transition rate (20) denotes the rate at which a tote moves to merge m_k if the tote does not need to visit the zone. If the zone is blocked ($i + y_{z_{k-1}} = q_{z_{k-1}} + 1$), the totes that are supposed to go to the zone are directly transported to the merge (21) with high priority. The rates for $\mathcal{H}^1$ are similarly defined, except (21) is not defined since the input buffer of the entrance station is assumed to be infinite.

The rate at which a tote leaves the zone or the merge is given as follows:

$$q_{(i,j,l)(i-1,j+1,l)} = \mu_{z_{k-1}}, \quad i > 0,\ j < l_{m_k} + 1, \tag{22}$$

$$q_{(i,j,0)(i,j-1,0)} = \mu^L_{m_k}, \quad j > 0, \tag{23}$$

$$q_{(i,j,l)(i,j,l-1)} = \mu^H_{m_k}, \quad l > 0. \tag{24}$$

Transition rate (22) denotes the rate of a service completion of a tote at zone z_{k-1}, whereas (23) and (24) denote the rate of a service completion of a tote at merge m_k for low and high priority totes, respectively. A low priority tote can only complete its service when there is no high priority tote at the merge. The rates for $\mathcal{H}^1$ are similarly defined, wherein (22) $\mu_{z_{k-1}}$ should be replaced with μ_e.

Figure 5 shows the Markov chain of subnetwork $\mathcal{H}^k$ on state space $\mathbb{W}^k(n)$ where the number of output buffer places equals $l_{m_i} = 1$. In the appendix, we show how the marginal queue-length probabilities $\pi_i^k(j \mid n)$, $n = 1, \ldots, N$, $i \in \mathcal{H}^k$ of the Markov chain can be obtained using matrix-geometric methods.

Given the marginal queue-length probabilities, the average throughput of the subnetwork with number of totes n is given by

$$X^k(n) = \sum_{j=1}^{n} \pi^k_{f_k}(j \mid n)\mu_{f_k}(j), \quad n = 1, \ldots, N, \tag{25}$$

which is used as the input to analyze subnetwork $\mathcal{H}^{k+1}$.

After analyzing the last subnetwork, the marginal queue-length probability for each node in the queueing network can be obtained by a disaggregation step using the marginal queue-length probability of the subnetworks. The marginal probability $\pi_i(j \mid l)$ of j totes present at node i given that the number of totes in the system is l can be obtained as follows:

$$\pi_i(j \mid l) = \begin{cases} \displaystyle\sum_{m_k=0}^{j} \pi_i^k(j \mid m_k) \prod_{p=k+1}^{W} \left[\sum_{m_p=0}^{j} \pi^p_{f_p}(m_{p-1} \mid m_p)\right] \\ \quad \cdot \pi^{W+1}_{f_W}(m_W \mid l), \quad i \in \mathcal{H}^k, \\ \pi_i^{W+1}(j \mid l), \quad i \in \mathcal{H}^{W+1}. \end{cases} \tag{26}$$

Using the marginal queue-length probabilities performance statistics, such as the system throughput, node utilization, queue lengths can now easily be calculated.

5.3. Algorithm

In this section, we summarize the approximation procedure for analyzing the queueing network of Section 4. As shown in Sections 5.1 and 5.2, we analyzed the queueing network by progressively aggregating parts of the network. However, in the first step of the algorithm, we calculate the visit ratios using the jump-over approximation and the unknown blocking probabilities of the zones b_i, $i \in \mathcal{Z}$, (see Section 4). To analyze the queueing network, we use a modified version of the approximative algorithm presented in Van der Gaast et al. (2012). Similar to the original algorithm, the blocking probabilities b_i, $i \in \mathcal{Z}$ are initialized by 0 and are subsequently updated until all of the differences between the current and the previous blocking probability are smaller than a small value ϵ. In each iteration of the current version of the algorithm, the blocking probabilities are obtained by analyzing the subnetworks $\mathcal{H}^k$, $k = 0, \ldots, W + 1$ for which we use the new analysis presented in Sections 5.1 and 5.2.

The approximation procedure can now be summarized as follows:

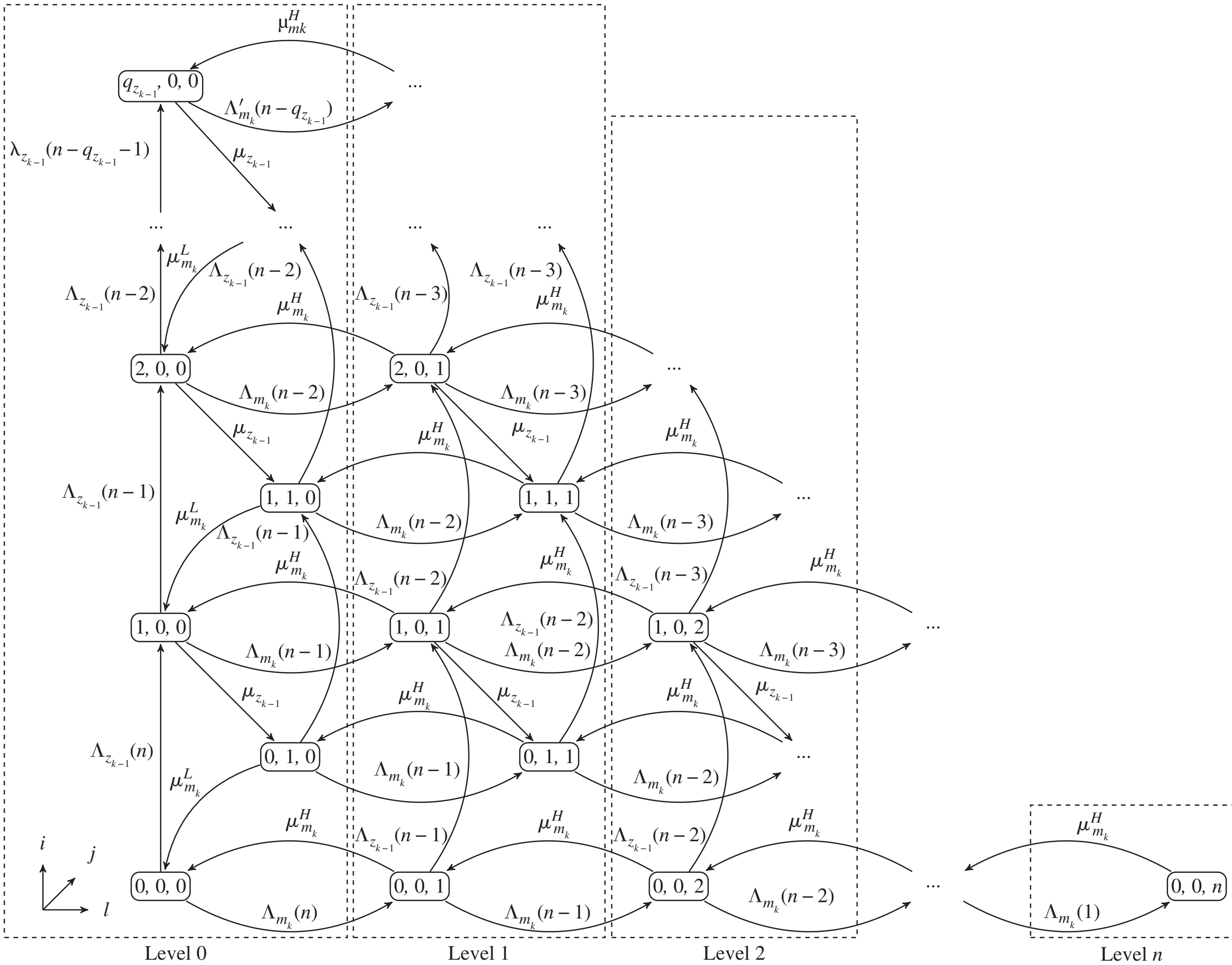
Step 1. Analyze the subnetwork $\mathcal{H}^0$ for different number of totes $n = 1, 2, \ldots, N$. For each n, obtain the marginal queue length probabilities $\pi_i^0(m \mid n)$, $i \in \mathcal{H}^0$ and throughput $X^0(n)$ (18).

Step 2. For $k = 1, \ldots, W + 1$. Construct FES f_{k-1} using (12) and analyze $\mathcal{H}^k$ for different number of totes $n = 1, 2, \ldots, N$. Obtain the marginal queue length probabilities $\pi_i^k(m \mid n)$ and throughput $X^k(n)$ (25).

Step 3. The throughput rate of the system is given by $X(N) = X^{W+1}(N)$ and the blocking probabilities are $b_i = \pi_i(q_i + 1 \mid N - 1)$, $i \in \mathcal{Z}$ (26).

Step 4. Go back to Step 1 with the new estimates for blocking probabilities b_i; continue until convergence up to ϵ of all of the blocking probabilities.

Figure 5. The Markov Chain of Subnetwork $\mathcal{H}^k$ on State Space $\mathbb{W}^k(n)$, $k = 2, \ldots, W+1$, with $l_{m_i} = 1$



Note. $\Lambda_{z_{k-1}}(n)$, $\Lambda_{m_k}(n)$, and $\Lambda'_{m_k}(n)$ are shorthand for (19)–(21), respectively.

6. Numerical Results

In this section, we compare the results of our approximation method with a discrete-event simulation of the real queueing network. We test the performance of the approximation method for a zone picking system without recirculation in Section 6.1 and test it with recirculation in Section 6.2. In Section 6.3, we analyze whether the order in which the subnetworks are analyzed in the approximation method has a significant effect on the performance statistics. Finally, in Section 6.4, we study the effect of additional places in the input or output buffer of a zone on the performance of the system.

For each run, 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 of the runs. In the algorithm $\epsilon = 10^{-3}$, and convergence usually occurs within a few iterations. All experiments are run on a Core i7 with 2.4 GHz and 8 GB RAM.

6.1. Zone Picking System Without Recirculation

To study the performance and accuracy of the algorithm of Section 5.3, we start by considering a zone picking system with 2, 4 or 6 zones without recirculation. The iterative algorithm of Van der Gaast et al. (2012) need not be used in case of no recirculation, since $b_i = 0$, $i \in \mathcal{Z}$. In a system with W zones, a tote can visit a total of 2^W possible combinations of zones. We assume that each combination of zones (a class) has the same probability of being released into the system, except the empty set, e.g., $\psi_\varnothing = 0$ and $\psi_\mathbf{r} = 1/(2^W - 1)$. Furthermore, we assume that each zone, merge, and conveyor is identical to a node of the same type. The time required to prepare a new tote to be launched into the system at the entrance station equals $\mu_e^{-1} = 5$ seconds. Each conveyor node requires a fixed deterministic time to cross of $\mu_i^{-1} = 60$ seconds, $i \in \mathcal{C}$, whereas the time

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 Without Recirculation

	2 zones				4 zones				6 zones			
N	Approx.	Simulation		Error	Approx.	Simulation		Error	Approx.	Simulation		Error
5	70.77	71.03	(±0.10)	−0.37	45.39	45.39	(±0.03)	−0.01	32.77	32.77	(±0.03)	0.02
10	119.34	120.49	(±0.32)	−0.95	86.20	86.23	(±0.08)	−0.04	63.47	63.50	(±0.07)	−0.05
15	141.93	142.91	(±0.40)	−0.69	120.84	120.83	(±0.15)	0.01	91.56	91.45	(±0.07)	0.12
20	150.14	150.82	(±0.40)	−0.45	148.03	148.19	(±0.20)	−0.11	116.49	116.54	(±0.10)	−0.04
25	153.70	153.89	(±0.32)	−0.13	167.62	167.69	(±0.42)	−0.04	137.84	137.86	(±0.15)	−0.01
30	155.64	155.94	(±0.22)	−0.19	180.88	181.07	(±0.36)	−0.11	155.41	155.49	(±0.29)	−0.05
35	156.87	157.12	(±0.34)	−0.16	189.69	189.91	(±0.40)	−0.12	169.36	169.24	(±0.34)	0.07
40	157.71	157.99	(±0.47)	−0.18	195.68	195.88	(±0.30)	−0.10	180.18	180.13	(±0.28)	0.02
45	158.33	158.66	(±0.38)	−0.21	199.91	199.95	(±0.52)	−0.02	188.47	188.67	(±0.35)	−0.11
50	158.80	158.83	(±0.43)	−0.02	203.02	202.86	(±0.65)	0.08	194.84	194.96	(±0.33)	−0.06
55	159.17	159.22	(±0.35)	−0.03	205.39	205.60	(±0.42)	−0.10	199.80	199.79	(±0.39)	0.00
60	159.47	159.47	(±0.42)	0.00	207.25	207.28	(±0.33)	−0.02	203.72	203.76	(±0.41)	−0.02

required to pass a merge node equals $(\mu_i^L)^{-1} = (\mu_i^H)^{-1} =$ 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 equals 1 and the input buffer sizes of each zone is, respectively, $q_i = \infty$, $i \in \mathcal{Z}$, which means that an incoming tote is always accepted by the buffer of the zone. Finally, we assume that there is no output buffer after a zone and the entrance ($l_i = 0,\ i \in \mathcal{M}$). The order picker or the entrance station can only begin work on the next tote in line when the current tote has crossed the merge.

Table 1 presents the results of the approximation method and the simulation method in terms of the average throughput $X(N)$ per hour for different numbers of totes in the system. The numbers in parentheses represent the standard deviations of the 10 different runs of the simulation model and the column *error* shows the relative error between the approximation and the simulation. The run time per case for the analytical model is less than a few seconds, whereas the simulation takes at most 30 seconds in the case of larger systems. The results show that the approximation method accurately predicts the average throughput of the system for each of the three configurations since all of the errors are within 1%. Also, for any N, the average throughput will never decrease due to the assumption of infinite input buffers for the zones.

6.2. Zone Picking System with Recirculation

For the next comparison, we test the performance and accuracy of the algorithm of Section 5.3 for a zone picking system with recirculation. We assume that all of the parameters are the same as in Section 6.2, with the exception that each zone now has a finite input buffer of size $q_i = 3$, $i \in \mathcal{Z}$.

Table 2 presents the results of the three configurations for the approximation method and the simulation in terms of the average throughput $X(N)$ per hour. Run times were similar to Section 6.1 for the simulation and the approximation. The approximation slightly overestimates the average throughput when reaching the maximum average throughput capability

Table 2. 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

of the system. For example, in the configuration with two zones, the maximum throughput capability that can reached is ±106 totes per hour if $N = 30$. Afterward the average throughput starts to decrease because totes flowing out of a zone must wait a long time until they can be merged on the main conveyor, thus preventing the order picker from continuing her work on the next tote in line and preventing the entrance station from releasing new totes. On the other hand, totes on the main conveyor recirculate until there is an open position in the input buffer of the zone. A similar effect can be seen in the configuration with four and six zones.

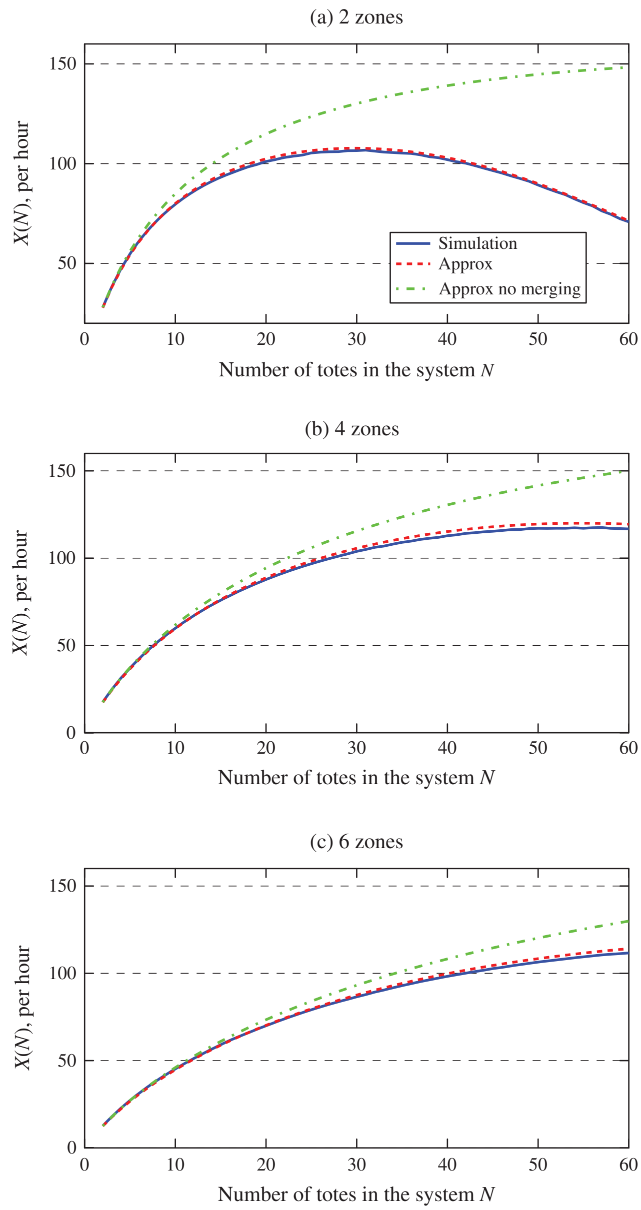
Figure 6 shows the same results for the average throughput, as well as the results of the approximation method where the merge node is replaced by a single-server queueing node with an infinite buffer and the same service distribution as the current merge node. This means that totes entering the merge are served on a first-come-first-served basis and the order picker/entrance station is never blocked because of a full output buffer. 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 when use of the order pickers equals 1. It can be concluded that modeling the merge operation in detail is of great importance, otherwise, the maximum throughput capability of the system cannot be correctly determined. This can lead to the expectation that the system has a much higher throughput capability than is possible in reality.

6.3. Order of Solving the Subnetworks

In the algorithm of Sections 5.3, 6.1, and 6.2, we assumed that the subnetworks were solved starting from the subnetwork with all of the conveyors, then the subnetwork with the entrance/exit station until the subnetwork with the last zone. Although any other sequence of analyzing the subnetwork is also feasible, based on Norton's theorem, this will not lead to the exact same results because the queueing network does not have a product-form solution. In this section, we test how the sequence of solving the subnetworks has an effect on the average throughput of the system.

We ran experiments for a zone picking system with $W = 4$ zones with a varying number of totes in the system $N = 2, \ldots, 40$. The input/output buffer sizes of the zones are assumed to be equal, and varied between 1, 2, and 3 positions. The other parameters are similar to Section 6.1. We test two extreme cases, i.e., solve the subnetworks starting with the conveyor subnetwork up to the subnetwork with the last zone (*forward*) and the reverse where the conveyor subnetwork is again analyzed first, but then the subnetwork with the last zone

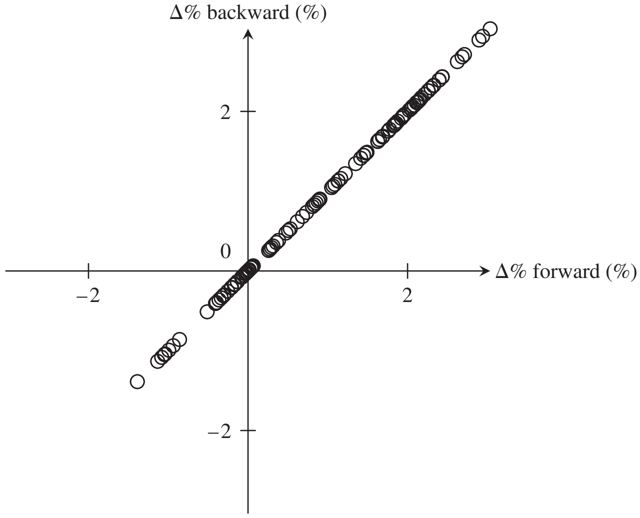
Figure 6. (Color online) Results of the Average Throughput $X(N)$ per Hour of the Approximation Model and the Simulation Model With and Without Merges Modeled for 2, 4, and 6 Zones Without Recirculation



until the subnetworks with the entrance/exit are analyzed (*backward*). In total, this gives 234 different cases (117/117 cases forward/backward).

Figure 7 shows the relative errors Δ% for the forward and backward method with simulation for the average throughput $X(N)$. Both approximations obtain results that are within 3%, compared to the simulation results. If we compare the results from the forward and backward method, we see that they fit almost perfectly on an increasing 45° line. This means that even when analyzing the subnetworks in a different sequence, similar results are obtained for the average throughput. This

Figure 7. Relative Errors in the Average Throughput $X(N)$ for Solving the Subnetworks in a Forward or Backward Sequence for a Zone Picking System with $W = 4$ Compared with the Simulation Model



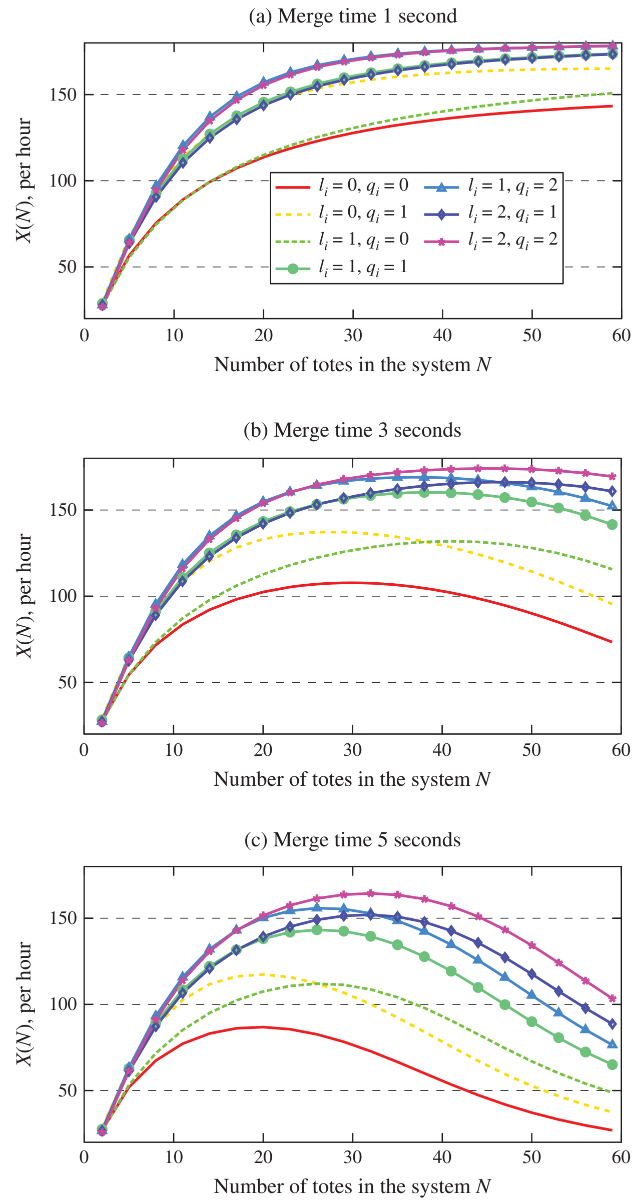
also holds for other node specific statistics such as the use and average queue lengths.

6.4. Effect of Buffer Sizes of the Zones

Finally, we test the effect of additional input and output positions in the buffer of the zones on the performance of a zone picking system with $W = 2$ zones. This is important for warehouse managers, since additional conveyor space is expensive and requires space. Therefore, deciding on the optimal number of input and output positions is essential for system performance, as well as for budget constraints. We assume that zones are parallel to the main conveyor (as shown in Figure 1) and that adding each additional buffer position increases the time required to travel the conveyor by three seconds. The number of positions in the input/output buffer of the zones, q_i and l_i, is varied from 0 to 2 and are assumed to be the same across all zones in the system. The other parameters are the same as in Section 6.2, except that the time required to pass a merge node $(\mu_i^L)^{-1} = (\mu_i^H)^{-1}$, $i \in \mathcal{M}$ varies between 1, 3, and 5 seconds.

Figure 8 presents the results for the average throughput for the three different merge times and the seven configurations of input/output buffer sizes. Only the results from the approximation model are shown, but the relative errors compared to the simulation model are of the same magnitude as in Section 6.2. If we compare Figures 8(a)–(c), we see that the maximum throughput capability increases as the merge times decrease when comparing the same input/output buffer positions. In addition, in Figure 8(c), the average throughput decreases much faster than in Figures 8(a) and 8(b).

Figure 8. (Color online) Results of the Average Throughput $X(N)$ per Hour of the Approximation Model and the Simulation Model for a Zone Picking System with Two Zones with Different Merging Times and Input/Output Buffer Sizes q_i and l_i



Also, in Figures 8(a)–(c), it can be seen that when N is low, it is more beneficial to have an additional position in the input buffer, since it decreases the possibility that a tote is rejected from entering the buffer of the zone and has to recirculate on the main conveyor. However, when N increases, it becomes more attractive to have an additional output buffer position, since the average time required to merge and the fact that the order picker is stopped more often becomes higher than the time it takes for a tote to recirculate once. Also, when the system is heavily used, the supply of new totes to the zones stalls due to congestion at the merges. Consequently, increasing the length of the output buffer is more attractive than the size of the input buffer. This can be especially seen when comparing $l_i = 0$, $q_i = 1$ with $l_i = 1$, $q_i = 0$, $i \in \mathcal{Z}$.

6.5. Real World Example

In this section, we test the approximation method using data from a real zone picking system at a large wholesaler supplying non-food items to supermarkets. The part of the warehouse dedicated to zone picking is divided into four interconnected segments. Each segment is a group of zones connected to a conveyor with a recirculation loop. The layout of a segment is similar to that shown in Figure 1, but with a different number of zones. The system entrance/exit of a segment now controls totes entering and leaving the segment. The first segment consists of three pallet picking zones, while the other three segments contain eight shelf picking zones each and use pick-by-light. In this example, we focus on the performance of the third segment. We consider this segment since customer totes that enter the system and visit this segment normally only visit this segment because it stores a distinct product range.

By analyzing the log files from the Warehouse Management System for several representative picking days, data about the average release probabilities and service times were obtained. Table 3 gives the parameters of the zones and the conveyor nodes. The mean picking times in the zones vary between 22 and 27 seconds, on average. In each zone, there is one order picker, while the input buffer size q_i, $i \in \mathcal{Z}$ is 8 or 11 depending on the location of the zone. Currently, order pickers must manually push the order totes back on the conveyor, which means that there is no output buffer after a zone; $l_i = 0$, $i \in \mathcal{Z}$. The time required to pass a merge node $(\mu_i^L)^{-1} = (\mu_i^H)^{-1}$, $i \in \mathcal{M}$ equals three seconds, on average. The time spent in the entrance station equals five seconds, on average. Finally, totes visit, on average, 3.22 zones.

In Table 4 we test the effect of substituting input buffer locations by output buffer locations on the average throughput $X(N)$. This means if $l_i = 1$ then q_i is reduced by one position. We vary the time required to pass a merge node $\mu_m^{-1} = (\mu_i^L)^{-1} = (\mu_i^H)^{-1}$, $i \in \mathcal{M}$ between 1, 3, and 5 seconds, and the maximum number of totes in the segment between 85, 95, and 105. Similar to the previous section we only present the results of the approximation, as the differences with the simulation are very small. From the results, we can see that when the merge time is 1 second $X(N)$ is similar as for N is 85, 95, and 105 and drops when the merge time increases. However, by increasing the output buffer space to one position, it can be seen that for all different merge times the average throughput $X(N)$ increases considerably. This is especially the case when the merge time equals five seconds. However, increasing the output buffer positions further by one additional position does not significantly increase $X(N)$. Therefore, the performance of the segment can benefit the most by substituting one input buffer location with one output buffer location. Overall, there is room to improve the current segment performance by decreasing the merge times or changing the input/output buffer locations.

Table 3. Overview Parameters Zones and Conveyor Nodes of the Real-World Case

Node	z_1^3	z_2^3	z_3^3	z_4^3	z_5^3	z_6^3	z_7^3	z_8^3	
μ_i^{-1} (sec)	25.2	26.7	26.0	25.3	22.2	21.9	23.4	25.4	
q_i	8	11	11	11	11	11	11	8	
l_i	0	0	0	0	0	0	0	0	
Node	c_1^3	c_2^3	c_3^3	c_4^3	c_5^3	c_6^3	c_7^3	c_8^3	c_9^3
μ_i^{-1} (sec)	37.0	37.0	37.0	37.0	62.0	37.0	37.0	37.0	25.0

Table 4. Results of the Average Throughput $X(N)$ per Hour of the Approximation Model for Varying Output Buffer Sizes l_i, Maximum Amount of Totes in the Segment, N, and Different Merge Times μ_m^{-1} (sec)

μ_m^{-1}		1 sec			3 sec			5 sec		
l_i	N	85	95	105	85	95	105	85	95	105
0		186.9	186.7	186.2	152.6	144.5	135.0	103.8	89.0	76.3
1		196.0	196.3	196.4	188.0	184.9	180.4	158.2	141.6	124.3
2		196.3	196.6	196.8	194.2	193.1	191.0	178.3	166.2	150.9

7. Conclusion and Further Research

In this paper, we developed an analytical model to study the merge operation in zone picking systems. We used a decomposition-based approximation to study each merge and zone in isolation. Our method progressively aggregated parts of the network and replaced the aggregated subnetwork with a flow equivalent single node. The approximation directly solved the global balance equations of the underlying Markov chain of the subnetworks for its steady-state distribution. The results show that for a wide range of parameters, the approximation predicted the maximum throughput capability of a zone picking system very accurately compared to simulation. The model is capable of predicting the loss in throughput given the level of congestion and blocking in the system, and can be used to allocate input and output buffer spaces to maximize the throughput capability of the system. Further research topics include studying other performance indicators, obtaining higher moments for the performance statistics, and investigating the performance of a zone picking system in an operational setting. Finally, our method to study conveyor merges in zone picking systems is quite generic and can also be applied to other internal logistic systems where merges of totes occur and can influence the throughput capability, such as sorting systems, vehicle-based AGV systems, and production lines with conveyors.

Appendix. Analysis of Subsystems $k \geq 1$

Figure 5 shows the Markov chain of subnetwork $\mathscr{H}^k$ on state space $\mathbb{W}^k(n)$ where the number of output buffer places equals $l_{m_i} = 1$. From Figure 5, we can see that the Markov chain is irreducible and that it is possible to partition state space $\mathbb{W}^k(n)$ such that

$$\mathbb{W}^k(n) = \bigcup_{l=0}^{n} \mathbb{W}_l^k(n), \quad \mathbb{W}_l^k(n) = \{w_{l,1}^k(n), \ldots, w_{l,s_l}^k(n)\}, \quad \mathbb{W}_l^k(n) \cap \mathbb{W}_{l'}^k(n) = \varnothing, \quad \text{for } l \neq l', \tag{27}$$

where partition $\mathbb{W}_l^k(n)$ consists of s_i states where the number of high priority totes at merge m_k equals l.

Then the states of $\mathbb{W}^k(n)$ can be arranged such that the generator matrix of (19)–(24) is of a block-tridiagonal form, that

$$Q = \begin{pmatrix} Q_{00} & Q_{01} & & & & \\ Q_{10} & Q_{11} & Q_{12} & & & \\ & Q_{21} & Q_{22} & Q_{23} & & \\ & & \ddots & \ddots & \ddots & \\ & & & Q_{n-1,n-2} & Q_{n-1,n-1} & Q_{n-1,n} \\ & & & & Q_{n,n-1} & Q_{nn} \end{pmatrix}, \tag{28}$$

where $Q_{ij} \in \mathbb{R}^{s_i \times s_j}$. In this case, the Markov process is said to be a finite level-dependent quasi-birth-and-death (LDQBD) process, and, $l = 1, \ldots, n$, the subset $\mathbb{W}_l^k(n)$ of states is referred to as the process level with level number l (Latouche and Ramaswami 1999). In Bright and Taylor (1995) an efficient procedure is given to compute the stationary distribution π of an LDQBD process. The procedure partitions stationary distribution $\pi = (\pi_0, \pi_1, \ldots, \pi_i, \ldots, \pi_n)$, with $\pi_i = (\pi^k(0,0,i), \pi^k(1,0,i), \pi^k(0,1,i), \ldots)$ and uses the fact that

$$\pi_{i+1} = \pi_i R_i, \quad i \geq 0, \tag{29}$$

with nonnegative matrices $R_i \in \mathbb{R}^{s_i \times s_{i+1}}$ that depend on the level. The basic idea behind the procedure is to exploit the fact that these matrices satisfy the infinite recurrence scheme

$$R_i = -Q_{i,i+1}(Q_{i+1,i+1} + R_{i+1}Q_{i+2,i+1})^{-1}, \tag{30}$$

and where π_0 is a solution of the equation

$$\pi_0(Q_{00} + R_0 Q_{10}) = 0. \tag{31}$$

Bright and Taylor (1995) show that the inverse matrices in (30) exist, and that $Q_{00} + R_0 Q_{10}$ has the characteristics of a generator matrix of an irreducible continuous time Markov chain with a finite state space. The solution of π_0 is unique up to a multiplicative constant and can be chosen to be nonnegative.

The boundary conditions at a maximum level n are given by

$$R_{n-1} = -Q_{n-1,n} Q_{nn}^{-1}. \tag{32}$$

Thus, an exact algorithm for computing the stationary distribution of finite LDQBDs proceeds as follows: Determine R_{n-1}, then compute R_i for $i = n-2, \ldots, 0$ using (30). Determine π_0 by choosing a nontrivial solution of (31), and use (29) for computing $\pi_1, \pi_2, \ldots, \pi_n$. Finally, normalize π.

After obtaining the marginal distribution of the Markov chain with transition matrix (28), it is now possible to calculate the marginal distribution of the nodes in subnetwork $\mathscr{H}^k$. Let $\pi_i^k(m \mid n)$ be the marginal distribution of node i in subnetwork $\mathscr{H}^k$ where there are m totes in the node and where the number of totes is n. The marginal distribution for the three nodes in the subnetwork is given as follows:

$$\pi_{f_{k-1}}^k(m \mid n) = \sum_{w \in \mathbb{W}^k(n):\ m = n-i-j-l} \pi^k(i,j,l), \tag{33}$$

$$\pi_{z_{k-1}}^k(m \mid n) = \sum_{w \in \mathbb{W}^k(n):\ m = i + y_{z_{k-1}}} \pi^k(i,j,l), \tag{34}$$

$$\pi_{m_k}^k(m \mid n) = \sum_{w \in \mathbb{W}^k(n):\ m = l+j-y_{z_{k-1}}} \pi^k(i,j,l). \tag{35}$$

References

Baskett F, Chandy KM, Muntz RR, Palacios FG (1975) Open, closed, and mixed networks of queues with different classes of customers. *J. ACM* 22(2):248–260.

Bastani AS (1988) Analytical solution of closed-loop conveyor systems with discrete and deterministic material flow. *Eur. J. Oper. Res.* 35(2):187–192.

Bastani AS, Elsayed EA (1986) Blocking in closed-loop conveyor systems connected in series with discrete and deterministic material flow. *Comput. Indust. Engrg.* 11(1):40–45.

Bolch G, Greiner S, de Meer H, Trivedi KS (2006) *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, 2nd ed. (Wiley-Interscience, Hoboken, NJ).

Boucherie RJ, van Dijk NM (1993) A generalization of Norton's theorem for queueing networks. *Queueing Systems* 13(1–3):251–289.

Bozer YA, Hsieh YJ (2005) Throughput performance analysis and machine layout for discrete-space closed-loop conveyors. *IIE Trans.* 37(1):77–89.

Bright L, Taylor PG (1995) Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes. *Stochastic Models* 11(3):497–525.

Bryant RM, Krzesinski AE, Lakshmi MS, Chandy KM (1984) The MVA priority approximation. *ACM Trans. Comput. Systems* 2(4): 335–359.

Buzen JP (1973) Computational algorithms for closed queueing networks with exponential servers. *Comm. ACM* 16(9):527–531.

Chandy KM, Herzog U, Woo L (1975) Parametric analysis of queuing networks. *IBM J. Res. Development* 19(1):36–42.

Chao X, Miyazawa M, Pinedo M (1999) *Queueing Networks: Customers, Signals and Product Form Solutions* (Wiley, Chichester, UK).

Coffman EG Jr, Gelenbe E, Gilbert EN (1988) Analysis of a conveyor queue in a flexible manufacturing system. *Eur. J. Oper. Res.* 35(3):382–392.

Dallery Y, Gershwin SB (1992) Manufacturing flow line systems: A review of models and analytical results. *Queueing Systems* 12(1): 3–94.

De Koster MBM (1994) Performance approximation of pick-to-belt orderpicking systems. *Eur. J. Oper. Res.* 72(3):558–573.

De Koster MBM, Le-Duc T, Roodbergen KJ (2007) Design and control of warehouse order picking: A literature review. *Eur. J. Oper. Res.* 182(2):481–501.

Disney RL (1962) Some multichannel queueing problems with ordered entry. *J. Indust. Engrg.* 13(1):46–48.

Economou A, Fakinos D (1998) Product form stationary distributions for queueing networks with blocking and rerouting. *Queueing Systems* 30(3):251–260.

Eisenstein DD (2008) Analysis and optimal design of discrete order picking technologies along a line. *Naval Res. Logist.* 55(4):350–362.

Gordon WJ, Newell GF (1967) Closed queuing systems with exponential servers. *Oper. Res.* 15(2):254–265.

Gu J, Goetschalckx M, McGinnis LF (2010) Research on warehouse design and performance evaluation: A comprehensive review. *Eur. J. Oper. Res.* 203(3):539–549.

Hsiao MT, Lazar AA (1989) An extension to Norton's equivalent. *Queueing Systems* 5(4):401–411.

Hsieh YJ, Bozer YA (2005) Analytical modeling of closed-loop conveyors with load recirculation. Gervasi O, Gavrilova ML, Kumar V, Lagana A, Lee HP, Mun Y, Taniar D, Tan CJK, eds. *Computational Science and Its Applications—ICCSA* 2005, Lecture Notes Comput. Sci., Vol. 3483 (Springer, Berlin Heidelberg), 437–447.

Jackson JR (1963) Jobshop-like queueing systems. *Management Sci.* 10(1):131–142.

Jane CC (2000) Storage location assignment in a distribution center. *Internat. J. Phys. Distribution Logist. Management* 30(1):55–71.

Jewkes E, Lee C, Vickson R (2004) Product location, allocation and server home base location for an order picking line with multiple servers. *Comput. Oper. Res.* 31(4):623–636.

Kritzinger PS, Van Wyk S, Krzesinski AE (1982) A generalisation of Norton's theorem for multiclass queueing networks. *Performance Evaluation* 2(2):98–107.

Kwo TT (1958) A theory of conveyors. *Management Sci.* 5(1):51–71.

Latouche G, Ramaswami V (1999) *Introduction to Matrix Analytic Methods in Stochastic Modeling*, Vol. 5 (SIAM, Philadelphia).

Malmborg CJ (1996) Storage assignment policy tradeoffs. *Internat. J. Production Res.* 34(2):363–378.

Marie RA (1979) An approximate analytical method for general queueing networks. *IEEE Trans. Software Engrg.* 5(5): 530–538.

Melacini M, Perotti S, Tumino A (2010) Development of a framework for pick-and-pass order picking system design. *Internat. J. Adv. Manufacturing Tech.* 53(9):841–854.

Muth EJ (1977) A model of a closed-loop conveyor with random material flow. *AIIE Trans.* 9(4):345–351.

Neuse D, Chandy KM (1982) HAM: The heuristic aggregation method for solving general closed queueing network models of computer systems. *ACM SIGMETRICS Performance Evaluation Rev.* 11(4):195–212.

Osorio C, Bierlaire M (2009) An analytic finite capacity queueing network model capturing the propagation of congestion and blocking. *Eur. J. Oper. Res.* 196(3):996–1007.

Park BC (2012) Order picking: Issues, systems and models. Manzini R, ed. *Warehousing in the Global Supply Chain: Advanced Models, Tools and Applications for Storage Systems*, 1st ed. (Springer-Verlag, London), 1–30.

Petersen CG (2000) An evaluation of order picking policies for mail order companies. *Production Oper. Management* 9(4):319–335.

Petersen CG (2002) Considerations in order picking zone configuration. *Internat. J. Oper. Production Management* 22(7):793–805.

Pittel B (1979) Closed exponential networks of queues with saturation: The Jackson-type stationary distribution and its asymptotic analysis. *Math. Oper. Res.* 4(4):357–378.

Reiser M, Lavenberg SS (1980) Mean-value analysis of closed multichain queuing networks. *J. ACM* 27(2):313–322.

Schassberger R (1984) Decomposable stochastic networks: Some observations. Baccelli F, Fayolle G, eds. *Modelling and Performance Evaluation Methodology*. Lecture Notes in Control and Information Sciences, Vol. 60 (Springer, Berlin), 135–150.

Schmidt LC, Jackman J (2000) Modeling recirculating conveyors with blocking. *Eur. J. Oper. Res.* 124(2):422–436.

Serfozo R (1999) *Introduction to Stochastic Networks* (Springer-Verlag, New York).

Sonderman D (1982) An analytical model for recirculating conveyors with stochastic inputs and outputs. *Internat. J. Production Res.* 20(5):591–605.

Van der Gaast JP, De Koster MBM, Adan IJBF, Resing JAC (2012) Modeling and performance analysis of sequential zone picking systems. Working paper, Erasmus University, Rotterdam, Netherlands. http://alexandria.tue.nl/repository/books/751517.pdf.

Van Dijk NM (1988) On Jackson's product form with "jump-over" blocking. *Oper. Res. Lett.* 7(5):233–235.

Van Dijk NM (1993) *Queueing Networks and Product Forms: A Systems Approach*, Vol. 4 (John Wiley & Sons, Chichester, UK).

Walrand J (1983) A note on Norton's theorem for queuing networks. *J. Appl. Probab.* 20(2):442–444.

Whitt W (1982) Approximating a point process by a renewal process, I: Two basic methods. *Oper. Res.* 30(1):125–147.

Yu M, De Koster MBM (2008) Performance approximation and design of pick-and-pass order picking systems. *IIE Trans.* 40(11): 1054–1069.

Zijm WHM, Adan IJBF, Buitenhek R, Van Houtum GJ (2000) Capacity analysis of an automated kit transportation system. *Ann. Oper. Res.* 93(1–4):423–446.