Operations Research

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

Capacity Analysis of Sequential Zone Picking Systems

Jelmer P. van der Gaast, René B. M. de Koster, Ivo J. B. F. Adan, Jacques A. C. Resing

To cite this article:

Full terms and conditions of use: https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article’s accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2019, INFORMS

Please scroll down for article—it is on subsequent pages

With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org
1. Introduction

Order picking, the process of picking products to fill customer orders, is the most labor-intensive warehouse activity and accounts for about 55% of warehouse total operating costs (Drury 1988). E-commerce and other recent trends in distribution and manufacturing have increased the importance of efficient order picking even more (Le-Duc and De Koster 2007). This paper focuses on the modeling and capacity analysis of sequential zone picking systems.

Zone picking is one of the most popular picker-to-parts order-picking methods, in which the order-picking area is zoned. In each zone, an order picker is responsible for picking from a dedicated part of the warehouse (Petersen 2002, Gu et al. 2010). In practice, the zones are often connected by conveyors to reduce travel. The major advantages of zone picking systems are high-throughput ability, scalability, and flexibility in handling both small and large order volumes and fit for use for different product sizes with a varying number of order pickers. These systems are often applied in warehouses handling customer orders with a large number of order lines and with a large number of different products kept in stock (Park 2012). A disadvantage of such systems, however, is congestion and blocking under heavy use, leading to long order throughput times. To reduce blocking and congestion, most systems use the block-and-recirculate protocol to dynamically manage workload. In this paper, the various elements of the system, such as conveyor lanes and pick zones, are modeled as a multiclass block-and-recirculate queueing network with capacity constraints on subnetworks. Because of this blocking protocol, the stationary distribution of the queueing network is highly intractable. We propose an approximation method based on jump-overflow blocking. Multiclass jump-overflow queueing networks admit a product-form stationary distribution and can be efficiently evaluated by mean value analysis and Norton’s theorem. This method can be applied during the design phase of sequential zone picking systems to determine the number of segments, number and length of zones, buffer capacities, and storage allocation of products to zones to meet performance targets. For a wide range of parameters, the results show that the relative error in the system throughput is typically less than 1% compared with simulation.

Abstract. This paper develops a capacity model for sequential zone picking systems. These systems are popular internal transport and order-picking systems because of their scalability, flexibility, high-throughput ability, and fit for use for a wide range of products and order profiles. The major disadvantage of such systems is congestion and blocking under heavy use, leading to long order throughput times. To reduce blocking and congestion, most systems use the block-and-recirculate protocol to dynamically manage workload. In this paper, the various elements of the system, such as conveyor lanes and pick zones, are modeled as a multiclass block-and-recirculate queueing network with capacity constraints on subnetworks. Because of this blocking protocol, the stationary distribution of the queueing network is highly intractable. We propose an approximation method based on jump-over blocking. Multiclass jump-over queueing networks admit a product-form stationary distribution and can be efficiently evaluated by mean value analysis and Norton’s theorem. This method can be applied during the design phase of sequential zone picking systems to determine the number of segments, number and length of zones, buffer capacities, and storage allocation of products to zones to meet performance targets. For a wide range of parameters, the results show that the relative error in the system throughput is typically less than 1% compared with simulation.
maintained, and no sorting and product consolidation is required (Petersen 2000).

There are two types of sequential zone picking systems: single-segment routing and multisegment routing. In single-segment routing, the conveyor forms a circular loop connecting all zones, whereas in multisegment routing, zones are grouped in segments, and within each segment, the zones are connected to a conveyor with a recirculation loop. The segments are then connected by a central (or main) conveyor that diverts the totes to the required segments (see Figure 1). Multisegment routing improves the system throughput significantly because of shorter conveyor loops that avoid unnecessary tote travel times. However, investment costs and space requirements are higher compared with single-segment routing.

De Koster (1994), Yu and De Koster (2008, 2009), and Melacini et al. (2010) model a zone picking system as a network of queues. They use Whitt’s queueing network analyzer (Whitt 1982) to estimate performance statistics, such as utilization, throughput rate of a zone, and mean and standard deviation of the throughput time of the totes. However, a crucial aspect not taken into consideration is blocking. In most environments, the workload of the zones exhibit variability because of differences in work profiles of the orders. In peak periods, zones can become congested, leading to blockages that can propagate throughout the network, resulting in starved zones and increased throughput times. This can affect the performance of a zone picking system significantly and should not be ignored. Identifying and quantifying the effect of blocking is crucial in the design of zone picking systems.

In a zone picking system, congestion occurs in zones as well as in segments. In both cases, the block-and-recirculate protocol is used to dynamically manage workload: a tote is blocked and recirculated on the conveyor loop if the destination buffer is full or the segment is congested. The tote potentially visits other zones or segments before attempting to reenter the zone or segment where it was blocked.

Queueing networks with the block-and-recirculate protocol are highly intractable: no exact results for the stationary distribution exist. In the literature, blocking protocols have been investigated for various applications (see Schmidt and Jackman (2000), Hsieh and Bozer (2005), and Osorio and Bierlaire (2009) for recent references on manufacturing systems with automated conveyors). For a review on queueing networks with blocking, the reader is referred to Papadopoulos et al. (1993), Perros (1994), and Balsamo et al. (2001). Yao and Buzacott (1987) were the first to study a variation of the block-and-recirculate protocol for flexible manufacturing systems. However, this variation is not applicable in the context of zone picking systems.

The objective of the paper is to develop a capacity model for sequential zone picking systems (hereafter zone picking) with either single-segment or multisegment routing, finite buffers, segment capacities, and the block-and-recirculate protocol. This model can be used in the design phase of zone picking systems to study the effects of layout, loading, and storage on blocking and congestion. It considerably extends the models of De Koster (1994), Yu and De Koster (2008, 2009), and Melacini et al. (2010) that consider zone picking systems with single-segment routing and no blocking. The capacity model is a multiclass queueing network with the block-and-recirculate protocol. Because an exact analysis of block-and-recirculate queueing networks is not feasible, we develop an approximation by iteratively estimating the blocking probabilities in a multiclass
queueing network with jump-over blocking (Van Dijk 1988, Economou and Fakinos 1998). We show that jump-over blocking admits a product-form stationary queue length distribution. Key to the approximation is to equip the jump-over queueing network with Markovian routing that reflects the relation to the block-and-recirculate queueing network; that is, the flows in both networks should match. It appears that the approximation is efficient and provides accurate estimates of key statistics. Hence, it is a powerful tool to support decisions on the required number of segments, number and size of zones, buffer capacities, or storage allocation of products to zones to meet performance targets.

This paper is organized as follows: Section 2 presents the model for single-segment routing zone picking systems and develops the approximation and analysis. The model is extended to multisegment routing zone picking systems in Section 3. In Section 4, we analyze the results of the approximation for a range of parameters via computational experiments. In the final section, we conclude and suggest model extensions and topics for further research.

2. Single-Segment Zone Picking Systems

A zone picking system with single-segment routing comprises three parts: the entrance/exit, the conveyors, and the zones. Figure 2(a) shows a single-segment zone picking system with two zones.

Order release is regulated by a workload control mechanism (Park 2012), which sets an upper bound on the number of totes in the zones and on the conveyor. When this bound is reached, the control mechanism only releases a new order when a tote with all required order lines leaves the system. This mechanism prevents the conveyor from becoming the bottleneck of the system. Once an order is released, it is assigned to a tote at the entrance station. The tote then receives its data, for example, its packing list, and moves to the buffer of a requested zone and enters if the buffer is not full. Otherwise, the tote is blocked and stays on the conveyor to potentially visit other zones before returning. When the picking process has finished, the picker pushes the tote back on the conveyor. The waiting time for an empty space on the conveyor is assumed to be negligible because of the workload control mechanism. The conveyor then transports the tote to the next zone. A weight check at the end of the conveyor loop ensures that the tote contains all the required order lines. Otherwise, the tote is sent back to the beginning of the loop and returns to the zones where it was blocked previously. When the tote has visited all the required zones, it exits the system, and a new tote is immediately released into the system. Hence, we assume a saturated system with an infinite supply of orders. This is a valid assumption for zone picking system design, which aims at studying the throughput capacity of the system. Operational issues, such as the effect of varying order arrival rates and order waiting times are not in the scope of this paper, but can be studied within the framework of semiopen queueing networks.

To model this system, we propose a queueing network, the topology of which is shown in Figure 2(b) for a case of two zones. The zone picking system is modeled as a closed queueing network with one entrance/exit, M zones, and M + 1 nodes describing the conveyor between either two adjacent zones or between the entrance/exit and the first or last zone. The system entrance/exit is denoted as e, \( \mathcal{Z} = \{z_1, \ldots, z_M\} \) denotes the set of zones, and \( \mathcal{C} = \{c_1, \ldots, c_{M+1}\} \) is the set of conveyors. Finally, \( \mathcal{I} = \{e\} \cup \mathcal{E} \cup \mathcal{Z} \) is the set of

Figure 2. (Color online) A Zone Picking System with Single-Segment Routing and Its Corresponding Queueing Network

A zone picking system with single-segment routing and two zones

Corresponding queueing network with system entrance/exit \( e \), conveyors \( \mathcal{C} = \{c_1, c_2, c_3\} \) and zones \( \mathcal{Z} = \{z_1, z_2\} \)
all nodes in the network. We make the following assumptions:

- There is an infinite supply of orders and totes at the entrance of the system. This implies that a leaving tote is always immediately replaced by a new tote. Each tote initially has class \( r \subseteq \mathbb{X} \); for example, \( r = \{z_2, z_3\} \) means that the tote has to visit the second and third zone. After visiting zone \( z_i \), its class changes to \( r \setminus \{z_i\} \), and when \( r = \emptyset \), all required zones have been visited.
- The total number of totes in the system is always \( N \). As long as the total number of totes in the zones and conveyor nodes is less than \( N \), new totes are released one by one from the entrance/exit at an exponential rate \( \mu_i \), reflecting the rate at which a tote is prepared to enter the system (i.e., unfolding, adding labels, printing the packing list, etc.).
- The conveyor nodes are delay nodes with an exponential delay with mean \( 1/\mu_i \), \( i \in \mathbb{X} \). Remark 1 discusses the extension to Erlang distributed and deterministic delays.
- Each zone \( i \in \mathbb{X} \) has \( d_i (\geq 1) \) servers, which represent the order pickers in the zone. Orders are picked in order of arrival. The order-picking time is exponentially distributed with rate \( \mu_i \), \( i \in \mathbb{X} \), which captures both variations in the pick time per tote and the queueing network with states \( x = (x_i : i \in \mathbb{X}) \) is a Markov process. Let \( S(N) \) be the state space of the network, that is, the set of states \( x \) for which \( \sum_{i \in \mathbb{X}} n_i = N \) and \( n_i \leq d_i + q_i, i \in \mathbb{X} \).

The routing of totes through the network proceeds as follows. A new tote of class \( r \subseteq \mathbb{X} \) is released at the system entrance with probability \( \psi_r \). These release probabilities correspond to a known order profile obtained from, for example, historical order data or forecasts. After release, a class \( r \) tote moves from the system entrance to the first conveyor node \( c_1 \). After conveyor node \( c_i \), the tote either enters the input buffer of zone \( i \) if \( z_i \in r \) and the buffer is not full or moves to the next conveyor node \( c_{i+1} \). If the tote needs to enter and the buffer is full, the tote skips the zone and moves to \( c_{i+1} \) while keeping the same class. If the buffer is not full, the tote enters the buffer of zone \( i \), and after waiting in the buffer, an order picker picks the required order lines. After all picks are completed, the tote enters \( c_{i+1} \), and its class changes to \( s = r \setminus \{z_i\} \).

After visiting the last conveyor node \( c_{M+1} \), all totes with \( r \neq \emptyset \) are routed to the first conveyor node \( c_1 \). The other totes move to the exit and are immediately replaced by a new tote waiting for release at the entrance.

Formally, let \( p_{\theta, s, r}(x) \) be the state-dependent routing probability that a class \( r \) tote is routed from node \( i \) to node \( j \), where it moves into the last position and changes to a class \( s \) tote given that the network is in state \( x \). Then

\[
\begin{align*}
p_{\theta, s, r}(x) &= \psi_r, \\
p_{c, i, r, s}(x) &= 1, \quad i = 1, \ldots, M, z_i \in r \text{ and } n_{z_i} < d_{z_i} + q_{z_i}, \\
p_{c, i, s, r}(x) &= 1, \quad i = 1, \ldots, M, z_i \notin r \text{ or } n_{z_i} = d_{z_i} + q_{z_i}, \\
p_{c, s, r, z_i}(x) &= 1, \quad i = 1, \ldots, M, z_i \in r, s = r \setminus \{z_i\}, \\
p_{c, i, 0, r}(x) &= 1, \\
p_{c, i, r, \emptyset}(x) &= 1, \quad r \neq \emptyset,
\end{align*}
\]

where the other routing probabilities are equal to zero.

The stationary distribution of this queueing network is intractable because of the finite buffers (Stidham 2002) and the block-and-recirculate protocol, which justifies the development of an approximation. The first step is to approximate the block-and-recirculate network by a jump-over network. This is described in Section 2.1. This jump-over network exhibits a product-form steady-state distribution as shown in Section 2.2. Closed-form formulas of the visit ratios are derived in Section 2.3. In Section 2.4, performance statistics of the jump-over network can be easily calculated by, for example, mean value analysis (MVA). Section 2.5 explains how the jump-over network is used to approximate the original network, and Section 2.6 investigates the quality of the approximation.

### 2.1. Jump-Over Network

We approximate the block-and-recirculate protocol by the jump-over blocking protocol (Van Dijk 1988). This protocol, also known as “overtake full stations, skipping, blocking and rerouting,” admits closed-form analytic results for single-class queueing networks (Pittel 1979, Schassberger 1984, Van Dijk 1988, Economou and Fakinos 1998). Figure 3 illustrates both blocking protocols.

In the block-and-recirculate protocol (Figure 3(a)), a class \( r \) tote that intends to visit zone \( z_i \in r \) either enters the input buffer of \( z_i \) or skips \( z_i \) if its input buffer is full. Class \( r \) totes skipping zone \( z_i \) maintain their class, and class \( r \) totes leaving \( z_i \) always change to class \( r \setminus \{z_i\} \) before entering conveyor \( c_{i+1} \) (because these totes visited \( z_i \)).

In the jump-over blocking protocol (Figure 3(b)), a class \( r \) tote also skips zone \( z_i \) if the input buffer is full and proceeds as a class \( r \) tote when leaving \( z_i \).
A Bernoulli trial then determines whether each class \( r \) tote leaving \( z_i \) changes class or not: it maintains class \( r \) with probability \( b_{zi} \) and changes to class \( s = r \setminus \{z_i\} \) otherwise independently of whether the tote visited or skipped zone \( z_i \). Hence, the routing probabilities under jump-over blocking are state independent and given by the following (cf. (2)–(4)): \[ \begin{align*}
    p_{ci,zi,r} & = 1, \quad i = 1, \ldots, M, z_i \in r, \\
p_{ci,zi,r} & = 1, \quad i = 1, \ldots, M, z_i \notin r, \\
p_{ci,zi,s} & = b_{zi}, \quad i = 1, \ldots, M, z_i \in r, \\
p_{ci,zi,s} & = 1 - b_{zi}, \quad i = 1, \ldots, M, z_i \in r, s = r \setminus \{z_i\},
\end{align*} \] where the other routing probabilities are the same as in the block-and-recirculate network.

Key to the approximation is to choose \( b_{zi} \) such that the flows of class \( r \) and \( r \setminus \{z_i\} \) totes entering \( c_{i+1} \) match under both protocols. This can be done by taking \( b_{zi} \) as the fraction of totes skipping \( z_i \) in the block-and-recirculate network. In other words, \( b_{zi} \) is the blocking probability of zone \( z_i \) under block and recirculate. Naturally, this blocking probability is not known but is estimated by an iterative algorithm in Section 2.5.

### 2.2. Product-Form Stationary Distribution

Let \( \lambda_{ir} \) be the visit ratio of a class \( r \) tote to node \( i \) satisfying the traffic equations
\[
\lambda_{ir} = \sum_{j \in \mathcal{S}} \sum_{s \in \mathcal{I}} \lambda_{is} p_{is,r,i} \mathcal{P}_{i,j,r}, \quad i \in \mathcal{I}, r \subseteq \mathcal{J}.
\] (11)

Equation (11) determines the visit ratios \( \lambda_{ir} \) up to a multiplicative constant. By substituting the routing probabilities (7)–(10), the traffic equations reduce to
\[
\begin{align*}
    \lambda_{ir} & = \lambda_{c_{i+1},i} \psi_r, \\
    \lambda_{c_i,r} & = \lambda_{c_{i+1},r} \psi_r + \lambda_{c_{i+1},r} r \subseteq \mathcal{J}, \\
    \lambda_{c_{i+1},r} & = \lambda_{c_{i+1},r} b_{zi}, \quad i = 1, \ldots, M, z_i \in r \subseteq \mathcal{J}, \\
    \lambda_{c_{i+1},r} & = \lambda_{c_{i+1},r} + \lambda_{c_{i+1},r}(1 - b_{zi}), \quad i = 1, \ldots, M, z_i \notin r \subseteq \mathcal{J}, \\
    \lambda_{c_{i+1},r} & = \lambda_{c_{i+1},r} + \lambda_{c_{i+1},r}(1 - b_{zi}), \quad i = 1, \ldots, M, z_i \in r \subseteq \mathcal{J},
\end{align*}
\] where the other visit ratios are equal to zero.

### Theorem 1

The single-segment jump-over network has stationary distribution
\[
\pi(x) = \frac{1}{G} \prod_{i \in \mathcal{E}, i = e} \pi_i(x_i),
\] (17)

where \( G \) is the normalization constant and \( \pi_i(x_i) \) are defined as
\[
\pi_i(x_i) = \begin{cases} 
\prod_{i=1}^{n_i} \frac{\lambda_i x_i^\mu_i}{\mu_i!}, & i \in \mathcal{E}, \\
\prod_{i=1}^{n_i} \frac{\lambda_i x_i^\mu_i}{\mu_i!} \frac{1}{\gamma(n_i)}, & i \in \mathcal{J}.
\end{cases}
\] (18)

with \( \gamma(n_i) = \left( \frac{n_i!}{d_i!} \right)^{n_i/d_i}, \) if \( n_i \leq d_i, \)
\( \gamma(n_i) = \left( \frac{n_i!}{d_i!} \right)^{n_i/d_i}, \) if \( n_i > d_i. \)

**Proof.** Let \( q(x,y) \) denote the transition rate from state \( x \) to \( y \). The transition rates are specified as follows. We use the notation \( x - r_{i,l} + s_{jl} \) to indicate the state obtained from \( x \) by removing the class \( r \) tote in position \( l \) of node \( i \) (so \( r = x_{il} \)) and inserting a class \( s \) tote into position \( k \) of node \( j \). The events in this network are (single) totes departing from one node and moving to the next one. If, in state \( x \), a new class \( r \) tote is released from entrance \( e \), it moves to conveyor \( c_1 \). The state after this event is \( x - e_{il} + r_{c_1,n_{i+1}}, \) and the rate is equal to
\[
q(x, x - e_{il} + r_{c_1,n_{i+1}}) = \mu_i \psi_r.
\] (19)

Naturally, this event is only feasible if \( n_i > 0 \). If the class \( r \) tote in position \( l \) of conveyor \( c_j \) (with \( i \leq M \)) completes transportation and \( z_i \notin r \), the tote continues to the next conveyor \( c_{i+1} \), so
\[
q(x, x - r_{c_{i+1}} + r_{c_{i+1},n_{i+1}}) = \mu_{c_{i+1}} z_i \notin r.
\] (20)

If \( z_i \in r \) and there is room in the buffer, the tote joins zone \( z_i \). Hence, if \( n_{zi} < d_{zi} + q_{zi} \),
\[
q(x, x - r_{c_{i+1}} + r_{c_{i+1},n_{i+1}}) = \mu_{c_{i+1}} z_i \in r.
\] (21)
and if \( n_{zi} = d_{zi} + q_{zi} \), the tote skips \( z_i \) and changes class to \( r \setminus \{z_i\} \) with probability \( 1 - b_{zi} \),

\[
q(x, x - r_{c1l} + s_{ zi, ni_{zi} + 1 } ) = \begin{cases} 
\mu_{zi} (1 - b_{zi}), & z_i \in r, s = r \setminus \{z_i\}, \\
\mu_{zi} b_{zi}, & z_i \in r, s = r.
\end{cases}
\]

(22)

If order picking for the class \( r \) tote in position \( l \) of zone \( z_i \) \((l \leq \min(d_{zi}, n_{zi}))\) finishes, the tote continues to \( c_{i+1} \).

\[
q(x, x - r_{c1l} + s_{ zi, ni_{zi} + 1 } ) = \begin{cases} 
\mu_{zi} (1 - b_{zi}), & s = r \setminus \{z_i\}, \\
\mu_{zi} b_{zi}, & s = r.
\end{cases}
\]

(23)

Finally, if the class \( r \) tote in position \( l \) of the last conveyor \( c_{M+1} \) completes transportation and \( r = \emptyset \), the tote joins exit \( e \), and otherwise, it continues to the first conveyor \( c_1 \). Hence,

\[
q(x, x - r_{cM+1l} + r_{j1l}) = \begin{cases} 
\mu_{cM+1l}, & r = \emptyset, j = e, \\
\mu_{cM+1l}, & r \neq \emptyset, j = c_1.
\end{cases}
\]

(24)

This completes the description of the nonzero transition rates. Note that the total rate from \( x \) is

\[
q(x) = \sum_{y \in \mathcal{S}(N)} q(x, y) = \mu_{I_{c_1}} I_{(n_{c_1} > 0)} + \sum_{i=1}^{M+1} n_{ci} \mu_{ci} + \sum_{i=1}^{M} \min(d_{zi}, n_{zi}) \mu_{zi},
\]

(25)

where \( I_{c_1} \) is the indicator function.

To prove (17) we use (Kelly 1979, theorem 1.13) stating that, if we can find a collection of numbers \( \tilde{q}(x, y), x, y \in \mathcal{S}(N) \) such that

\[
\tilde{q}(x) = q(x), \quad x \in \mathcal{S}(N),
\]

(26)

and a collection of positive numbers \( \pi(x), x \in \mathcal{S}(N) \), summing to unity, such that

\[
\pi(x) \tilde{q}(x, y) = \pi(y) \tilde{q}(y, x), \quad x, y \in \mathcal{S}(N),
\]

(27)

then \( \tilde{q}(x, y) \) are the transition rates of the time-reversed process and \( \pi(x) \) is the stationary distribution of both processes. Obviously, the proposed collection \( \pi(x) \) is (17). Equation (27) then defines the rates \( \tilde{q}(x, y) \), and the only thing that remains to be done is verifying (26).

First, we determine the rates \( \tilde{q}(x, y) \) from (27).

At entrance \( e \), only class \( \emptyset \) totes arrive from conveyor \( c_{M+1} \). Hence, from (24), (27) and (17), we get, for \( l = 1, \ldots, n_{c_{M+1}} + 1 \),

\[
\tilde{q}(x, x - e_{c1l} + e_{c1l}) = \mu_{cM+1l} \frac{\pi(x - e_{c1l} + e_{c1l})}{\pi(x)}
\]

\[
= \mu_{cM+1l} \frac{1}{\left. \frac{\lambda_{c1l}}{\lambda_{e}} \right|_{\lambda_{e}}}
\]

\[
= \frac{\mu_{cM+1l}}{\lambda_{e}} \frac{\lambda_{c1l}}{n_{c_{M+1}} + 1}.
\]

(28)

provided \( n_e > 0 \). In conveyor \( c_{i+1} \), class \( r \) totes arrive from entrance \( e \) and conveyor \( c_{M+1} \), so by (19),

\[
\tilde{q}(x, x - r_{c1l} + \emptyset_{c1l}) = \frac{\pi(x - r_{c1l} + \emptyset_{c1l})}{\pi(x)}
\]

\[
= \frac{n_{ci} \mu_{ci}}{\lambda_{c1r} \lambda_{c1r}} \frac{\lambda_{c1r}}{n_{c_{M+1}} + 1}.
\]

(29)

and by (24), for \( l = 1, \ldots, n_{c_{M+1}} + 1 \),

\[
\tilde{q}(x, x - r_{c1l} + r_{c1l}) = \frac{n_{ci} \mu_{ci}}{\lambda_{c1r} \lambda_{c1r}} \frac{\lambda_{c1r}}{n_{c_{M+1}} + 1}.
\]

(30)

In conveyor \( c_{i+1} \), class \( r \) totes arrive from \( c_1 \) and \( z_i \). If \( z_i \notin r \), then by (20), for \( l = 1, \ldots, n_{c_{i+1}} + 1 \),

\[
\tilde{q}(x, x - r_{zi} + r_{c1l}) = \frac{n_{ci} \mu_{ci}}{\lambda_{c1r} \lambda_{c1r}} \frac{\lambda_{c1r}}{n_{c_{i+1}} + 1}.
\]

(31)

If \( n_{zi} = d_{zi} + q_{zi} \), then class \( r \) totes also arrive from \( c_1 \) by changing class. By (22), for \( l = 1, \ldots, n_{c_{i+1}} + 1 \),

\[
\tilde{q}(x, x - r_{zi} + r_{c1l}) = \frac{n_{ci} \mu_{ci}}{\lambda_{c1r} \lambda_{c1r}} \frac{\lambda_{c1r}}{n_{c_{i+1}} + 1}.
\]

(32)

Otherwise, if \( n_{zi} < d_{zi} + q_{zi} \), class \( r \) totes arrive from \( z_i \). By (23), for \( l = 1, \ldots, \min(d_{zi}, n_{zi} + 1) \),

\[
\tilde{q}(x, x - r_{zi} + r_{c1l}) = \frac{n_{ci} \mu_{ci}}{\lambda_{c1r} \lambda_{c1r}} \frac{\lambda_{c1r}}{n_{c_{i+1}} + 1}.
\]

(33)

Also class \( r \) totes with \( z_i \notin r \) arrive in \( c_{i+1} \) from \( c_i \) and \( z_i \). If \( z_i \in r \), then by (22), for \( l = 1, \ldots, n_{c_{i+1}} + 1 \),

\[
\tilde{q}(x, x - r_{zi} + r_{c1l}) = \frac{n_{ci} \mu_{ci}}{\lambda_{c1r} \lambda_{c1r}} \frac{\lambda_{c1r}}{n_{c_{i+1}} + 1}.
\]

(34)

if \( n_{zi} = d_{zi} + q_{zi} \), and otherwise, if \( n_{zi} < d_{zi} + q_{zi} \), then by (23), for \( l = 1, \ldots, \min(d_{zi}, n_{zi} + 1) \),

\[
\tilde{q}(x, x - r_{zi} + r_{c1l}) = \frac{n_{ci} \mu_{ci}}{\lambda_{c1r} \lambda_{c1r}} \frac{\lambda_{c1r}}{n_{c_{i+1}} + 1}.
\]

(35)

Finally, in zone \( z_i \), class \( r \) totes arrive from \( c_i \) with \( z_i \in r \). By (21), for \( l = 1, \ldots, n_{c_{i+1}} + 1 \),

\[
\tilde{q}(x, x - r_{zi} + r_{c1l}) = \frac{\min(d_{zi}, n_{zi} + 1) \mu_{zi}}{\lambda_{c1r} \lambda_{c1r}} \frac{\lambda_{c1r}}{n_{c_{i+1}} + 1}. \]

(36)

This completes the specification of the nonzero rates \( \tilde{q}(x, y) \).
To determine $\tilde{q}(x)$, note that, by (28), the rate of events in $x$ coming from entrance $e$ is equal to
\[
\sum_{j=1}^{n_{ci}^{++}} \tilde{q}(x, x - r_{c_{ij}, 1} + \theta_{c_{ij}, 1} + \theta_{c_{ij}, 1}^\theta ) = \sum_{j=1}^{n_{ci}^{++}} \frac{\mu_e \lambda_{c_{ij}, 1}^\theta}{\lambda_{c_{ij}, 1}} n_{ci} 1 + 1
= \mu_e \lambda_{c_{ij}, 1}^\theta \frac{\lambda_{c_{ij}, 1}^\theta}{\lambda_{c_{ij}, 1}^\theta} = \mu_e,
\]
provided $n_{e} > 0$. The last equality follows from the traffic Equation (12). By (29) and (30) and using traffic Equation (13), it follows that the rate of events resulting from $c_{1}$ is given by
\[
\tilde{q}(x, x - r_{c_{1}, n_{c_{1}, 1} + \theta_{c_{1}, 1} + \theta_{c_{1}, 1}^\theta}) = n_{c_{1}} \mu_{c_{1}} \lambda_{r_{c_{1}, 1}^\theta} + \lambda_{c_{1}, r_{c_{1}}} = n_{c_{1}} \mu_{c_{1}},
\]
if $z_{1} \notin r$, then it follows from (31)–(33) that the rate of events resulting from $c_{1}^{++}$ is
\[
\sum_{l=1}^{n_{c_{1}, 1}^{++}} \tilde{q}(x, x - r_{c_{1}, l_{n_{c_{1}, 1}} + \theta_{c_{1}, 1}}) + \sum_{l=1}^{n_{c_{1}, 1}^{++}} \tilde{q}(x, x - r_{c_{1}, l_{n_{c_{1}, 1}} + r \cup \{z_{1}\}}) = n_{c_{1}, 1} \mu_{c_{1}} \lambda_{r_{c_{1}, 1}^\theta} + \lambda_{c_{1}, r_{c_{1}}} = n_{c_{1}, 1} \mu_{c_{1}},
\]
and otherwise, if $z_{2} < d_{2} + q_{2}$,
\[
\sum_{l=1}^{n_{c_{1}, 1}^{++}} \tilde{q}(x, x - r_{c_{1}, l_{n_{c_{1}, 1}} + \theta_{c_{1}, 1}}) + \sum_{l=1}^{n_{c_{1}, 1}^{++}} \tilde{q}(x, x - r_{c_{1}, l_{n_{c_{1}, 1}} + r \cup \{z_{1}\}}) = n_{c_{1}, 1} \mu_{c_{1}} \lambda_{r_{c_{1}, 1}^\theta} + \lambda_{c_{1}, r_{c_{1}}} = n_{c_{1}, 1} \mu_{c_{1}},
\]
where the last equalities in (37) and (38) follow from traffic Equations (15) and (16). Alternatively, if $z_{1} \in r$, then it follows from (34) and (35) and traffic Equations (14) and (16), that the rate of events resulting from $c_{1}^{++}$ is also equal to $n_{c_{1}, 1} \mu_{c_{1}}$. By (36) and (16), the rate of events resulting from $z_{1}$ is
\[
\sum_{l=1}^{n_{c_{1}, 1}^{++}} \tilde{q}(x, x - r_{c_{1}, l_{n_{c_{1}, 1}} + \theta_{c_{1}, 1}}) = \min(d_{z_{1}, n_{z_{1}, 1}}),
\]
and the total rate $\tilde{q}(x)$ is now obtained by adding the rate of events resulting from each of the nodes in the network. It is exactly the same as $q(x)$ given by (25). \qed

Theorem 1 provides the stationary distribution of the jump-over network with detailed states $x$. However, knowledge of the number of totes in each node is sufficient to determine statistics, such as throughput and waiting times in the zones. To present the stationary distribution in terms of the aggregate vector $(n_{i} : i \notin \mathcal{F})$, we transform the class-dependent visit ratios $\lambda_{i, r}$ into chain visit ratios
\[
\nu_{i} = \frac{\sum_{r \in \mathcal{F}} \lambda_{i, r}}{\sum_{r \in \mathcal{F}} \lambda_{r}} , \quad i \notin \mathcal{F},
\]
which can be interpreted as the average number of times an arbitrary tote visits node $i$ before moving to the exit node $e$. Because every tote can be replaced by one of any other class at the exit, the jump-over network has a single chain of classes with a population of $N$ totes.

**Corollary 1.** In aggregate form, the stationary distribution of the jump-over network is
\[
\pi(n_{i} : i \notin \mathcal{F}) = \frac{1}{G} \prod_{i \notin \mathcal{F}} \pi_{i}(n_{i}),
\]
where $G$ is the normalization constant and $\pi_{i}(n_{i})$ are defined as
\[
\pi_{i}(n_{i}) = \begin{cases} \left( \frac{V_{i}}{\mu_{i}} \right)^{n_{i}}, & i = e, \\ \left( \frac{V_{i}}{\mu_{i}} \right)^{n_{i}} \frac{1}{\gamma(n_{i})}, & i \notin \mathcal{F}, \end{cases}
\]
where $\gamma(n_{i})$ is still $1/\mu_{i}$, though its variability is less than exponential. This larger network also possesses a stationary distribution of the form (17) and (40), in which the set of conveyors is extended to $\{c_{1}, \ldots, c_{k} : i = 1, \ldots, M + 1\}$ and in which $c_{1}, \ldots, c_{k}$ have the same visit ratio as $c_{i}$. By considering the total number of totes $n_{i}$ in the tandem $c_{1}, \ldots, c_{k}$, we obtain
\[
\sum_{n_{1} + \cdots + n_{k} = n_{i}} \pi_{c_{1}}(n_{1}) \cdots \pi_{c_{k}}(n_{k}) = \frac{1}{n_{i}!} \sum_{n_{1} + \cdots + n_{k} = n_{i}} \frac{V_{c_{1}}^{n_{1}}}{(k \mu_{c_{1}})^{n_{1}}} \cdots \frac{V_{c_{k}}^{n_{k}}}{(k \mu_{c_{k}})^{n_{k}}}
\]
Hence, the aggregate product-form (40) is also valid if the delay in conveyor node $c_{i}$ is Erlang-$k$ distributed with mean $1/\mu_{c_{i}}$. Because this is true for every $k$, we
can take $k$ to $\infty$ to conclude that Corollary 1 remains valid for conveyor nodes $c_i$ with constant delay $1/\mu_i$ (Kelly 1979, lemma 3.9).

**Remark 2.** The notion of quasi-reversibility by Kelly (1979) was extended by Chao and Miyazawa (2000) and Henderson and Taylor (2001). They show that quasi-reversibility can be applied to obtain product-form solutions for queueing networks with signals, negative customers, transitions involving three or more nodes, and batch movements. This approach can also be used to study the jump-over network.

### 2.3. Chain Visit Ratios

To obtain the chain visit ratios $V_i$, $i \in \mathcal{I}$, we need to solve the traffic equation (11). This might, however, require a large computational effort because the number of tote classes, $2^M$, grows exponentially with the number of zones. Alternatively, $V_i$ can be calculated directly per node type, that is, entrance/exit, conveyor, and zone. Clearly, $V_e = 1$ by (39). In the next subsections, we explain how the chain visit ratios of conveyors and zones are calculated.

#### 2.3.1. Conveyor Nodes

A tote visits all conveyor nodes with the same frequency during its stay in the system. As a result, the chain visit ratios $V_i$ of conveyor nodes are all the same and equal the average number of circulations of an arbitrary tote in the system before moving to the exit.

Let $X_i$ denote the class of a tote at the last conveyor $c_{i+1}$ after its $i$th circulation. Then $\{X_i, l \geq 0\}$ is an absorbing Markov chain with state space consisting of all subsets of $\mathcal{I}$, transition probability matrix $\Phi$, and absorbing state $\emptyset$. The chain starts in state $X_0 = r$ with probability $\psi_r$. The average number of circulations is then equal to the expected number of transitions before entering the absorbing state $\emptyset$. Transition probability $\Phi_{rs}$ from state $r$ at the start of a circulation to $s$ at the end of this circulation is given by the following (cf. (8)–(10)):

$$
\Phi_{rs} = \prod_{j \in s} b_j \prod_{i \in r \setminus s} (1 - b_i), \quad s \subseteq r \subseteq \mathcal{I},
$$

and zero otherwise. Note that transitions are only possible to states with fewer zones. Hence, the states can be ordered such that $\Phi$ is an upper triangular matrix, which can be written as

$$
\Phi = \begin{bmatrix}
\Theta & \Upsilon \\
0 & 1
\end{bmatrix},
$$

where $\Theta$ is an upper triangular matrix of transition probabilities between transient states, and $\Upsilon$ is a column vector of transition probabilities from transient states to absorbing state $\emptyset$. The last row of $\Phi$ corresponds to state $\emptyset$. The expected number of transitions until absorption is then given by (Wolff 1989)

$$
V_i = \psi(I - \Theta)^{-1}, \quad i \in \mathcal{C},
$$

where $I$ is the identity matrix, $1$ the column vector with ones, and $\psi = (\psi_r : r \subseteq \mathcal{I}\setminus\emptyset)$ the row vector with release probabilities ordered in the same way as $\Theta$. Because $(I - \Theta)$ is an upper triangular, its inverse can be easily determined by back-substitution. Denote $\omega = (I - \Theta)^{-1}$; then the $j$th element of $\omega$ follows from the recursion

$$
\omega_j = \left(1 + \sum_{k=j+1}^{2^M-1} \Theta_{jk}\omega_k\right)/(1 - \Theta_{jj}), \quad j = 2^M - 1, 2^M - 2, \ldots, 1.
$$

#### 2.3.2. Zones

The chain visit ratios $V_i$ of the zones are equal to the mean number of times an arbitrary tote visits zone $i$ before leaving the system. In the jump-over network, the number of times a tote visits zone $i$ follows a geometric distribution with success probability $1 - b_i$. Hence,

$$
V_i = \sum_{r : r \subseteq i} \frac{\psi_r}{1 - b_i}, \quad i \in \mathcal{I}.
$$

### 2.4. Mean Value Analysis

We formulate an MVA algorithm (Reiser and Lavenberg 1980) to efficiently compute key performance statistics of the jump-over network. MVA is based on the arrival theorem, which holds because of the product form solution of the jump-over network. Let $E(T_i(n))$ be the expected time a tote spends in node $i$ per visit, $X(n)$ be the system throughput, $E(L_i(n))$ be the mean number of totes in node $i$, and $\pi_i(j|n)$ be the marginal queue length probabilities of $j$ totes in zone $i$ given there are $n$ totes in the network. MVA then iteratively calculates these statistics.

First, initialize $E(L_i(0)) = 0$, $i \in \mathcal{I}$, and $\pi_i(0|0) = 1$, $\pi_i(j|0) = 0$ for $j = 1, \ldots, d_i + q_i$, if $i \in \mathcal{I}$. The mean throughput time $E(T_i(n))$ in entrance/exit $e$ and conveyor nodes $i \in \mathcal{C}$ can be calculated by

$$
E(T_i(n)) = \begin{cases}
\frac{1}{\mu_i} & (1 + E(L_i(n-1))), \quad \text{if } i = e, \\
\frac{1}{\mu_i} & \text{if } i \in \mathcal{C}.
\end{cases}
$$

This follows from the arrival theorem and the fact that $e$ is a single server and conveyor nodes $i \in \mathcal{C}$ are infinite servers. The mean throughput time in zones $i \in \mathcal{I}$ can be calculated by

$$
E(T_i(n)) = \sum_{j=d_i}^{d_i+q_i-1} \left(1 + 1 - d_i\right) \frac{1}{d_i \mu_i} \pi_i(j|n-1) + \frac{1}{\mu_i} \{1 - \pi_i(d_i + q_i|n-1)\}, \quad i \in \mathcal{I}.
$$
The first term of (46) is the average waiting time given the number of totes \( j \) in zone \( i \) on arrival, and the second term is the tote’s own average service time. When the buffer of zone \( i \) is full, the throughput time is zero because the tote skips the zone. The system throughput \( X(n) \) is given by (Reiser and Lavenberg 1980)

\[
X(n) = \sum_{i \in S} V_i E(T_i(n)),
\]

where the denominator is the average time a tote spends in the system, that is, the system throughput time. Applying Little’s law yields

\[
E(L_i(n)) = V_i X(n) E(T_i(n)), \quad i \in S.
\] (48)

Finally, we determine the marginal queue length probabilities by balancing the number of transitions per time unit between state \( j - 1 \) and \( j \), where \( j \) is the number of totes in zone \( i \). The rate from \( j \) to \( j - 1 \) is given by \( \min(j, d_i) \mu_i \pi_i(j) \) and, by the arrival theorem, the rate from \( j - 1 \) to \( j \) is \( V_i X(n) \pi_i(j - 1) \). Hence,

\[
\pi_i(j) = \frac{V_i X(n)}{\mu_i \min(j, d_i)} \pi_i(j - 1) \quad , \quad j = 1, \ldots, d_i + q_i, i \in S,
\]

where \( \pi_i(0) \) follows from normalization

\[
\pi_i(0) = 1 - \sum_{j=1}^{d_i+q_i} \pi_i(j), \quad i \in S.
\] (50)

Equation (50) has often been reported as the cause of numerical instability in MVA (Chandy and Sauer 1980). A numerically stable alternative is equation (27) of Reiser (1981).

Performance statistics for \( n = N \) can be determined by subsequently applying (45)–(50) for \( n = 1, \ldots, N \). Key statistics are the system throughput time \( \sum_{i \in S} V_i E(T_i(n)) \); the probability that an arriving tote in zone \( i \) is blocked, that is, \( b_i = \pi_i(d_i + q_i|N - 1) \); and the utilization \( \rho_i \) given by

\[
\rho_i = \begin{align*}
\frac{X(N)}{\mu_i}, & \quad \text{if } i = e, \\
\frac{V_i X(N)}{\mu_i}, & \quad \text{if } i \in \mathcal{E}, \\
1 - \sum_{j=0}^{d_i-1} \left| \frac{(d_i - j)}{d_i} \right| \pi_i(j|N), & \quad \text{if } i \in S.
\end{align*}
\] (51)

where \( \rho_e \) is the fraction of time the entrance/exit is busy; \( \rho_i, i \in \mathcal{S} \), is the fraction of time a picker in zone \( i \) is busy; and \( \rho_i, i \in \mathcal{E} \) is the average number of totes in conveyor node \( i \).

Remark 3. MVA is also exact in a jump-over network with deterministic conveyor delays (cf. Remark 1), but it is no longer exact in a network with nonexponential queuing times. Still, closed queuing networks are known to be robust to service distributions (Bolch et al. 2006). When zone \( i \) is full, an arriving tote skips zone \( i \) and its throughput time is zero. Otherwise, the tote enters zone \( i \), where it first has to wait for the first departure in zone \( i \) and then continues to wait for as many departures as there are totes waiting for its arrival before entering service. Hence, by adopting the arrival theorem as an approximation (Adan and Van der Wal 2011), we get

\[
E(T_i(n)) = Q_i(n - 1) \frac{E(R_i)}{d_i} + \sum_{j=d_i+q_i} E(B_i) \pi_i(j|n - 1) + E(B_i)(1 - \pi_i(d_i + q_i|n - 1)), \quad i \in S,
\] (52)

which replaces (46). In (52), \( E(B_i) \) is the expected service time of zone \( i \), \( E(R_i) = E(B_i^2)/(2E(B_i)) \) is the expected residual service time of zone \( i \), and \( Q_i(n - 1) = \sum_{j=d_i+q_i} \pi_i(j|n - 1) \) is the probability that all order pickers are busy upon entering zone \( i \). Queue length probabilities \( \pi_i(j|n) \) can again be determined by balancing the number of transitions per time unit between state \( j - 1 \) and \( j \), assuming as approximation that each order picker has an exponential service rate \( 1/E(B_i) \).

2.5. Iterative Algorithm for Calculating Blocking Probabilities

In the jump-over network, totes leaving zone \( i \) randomly change class according to probability \( b_i \), independent of the state of zone \( i \), \( i \in \mathcal{S} \). Probability \( b_i \) should match the blocking probability of zone \( i \) in the block-and-recirculate network. However, this blocking probability is not known in advance, but can be iteratively estimated. Initially, we set \( b_i^{(1)} = 0, i \in \mathcal{S} \). Then, we calculate the marginal queue length probabilities using Equations (49) and (50) and take the fraction of totes finding on arrival \( d_i + q_i \) totes in zone \( i \) as new estimate for \( b_i \). Thus, by the arrival theorem,

\[
b_i^{(m+1)} = \pi_i^{(m)}(d_i + q_i|N - 1), \quad i \in \mathcal{S},
\] (53)

where superscript \( (m) \) indicates that the quantity has been calculated in the \( m \)th iteration. Based on this new estimate for \( b_i \), routing probabilities and chain visit ratios are recalculated for all zones and conveyor nodes. After applying MVA, Equation (53) is used again to get a better estimate of \( b_i \) and so on. This process continues until the difference between \( b_i^{(m+1)} \) and \( b_i^{(m)} \) for all \( i \) is less than some small \( \epsilon \). The final estimates for performance statistics are then calculated. In our experience, convergence is fast and does not depend on the initial estimates of \( b_i \).

2.6. Example of the Single-Segment Routing Model

To illustrate the performance and accuracy of the jump-over approximation, we consider the zone picking
### Table 1. Performance Statistics for the Example with Varying Number of Totes N

<table>
<thead>
<tr>
<th>N</th>
<th>X(N) (in hours⁻¹)</th>
<th>Error, %</th>
<th>E(T_f(N)) (in seconds)</th>
<th>Error, %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sim</td>
<td>Jump</td>
<td>CQN</td>
<td>YdK</td>
</tr>
<tr>
<td>10</td>
<td>104.4 ±0.16</td>
<td>104.5</td>
<td>108.2</td>
<td>107.9</td>
</tr>
<tr>
<td>20</td>
<td>182.8 ±0.25</td>
<td>182.9</td>
<td>206.5</td>
<td>204.9</td>
</tr>
<tr>
<td>30</td>
<td>234.3 ±0.52</td>
<td>235.3</td>
<td>283.5</td>
<td>276.2</td>
</tr>
<tr>
<td>40</td>
<td>268.8 ±0.58</td>
<td>269.8</td>
<td>326.2</td>
<td>313.4</td>
</tr>
<tr>
<td>50</td>
<td>291.5 ±0.39</td>
<td>293.0</td>
<td>342.0</td>
<td>330.1</td>
</tr>
<tr>
<td>100</td>
<td>336.4 ±0.65</td>
<td>338.6</td>
<td>354.9</td>
<td>350.1</td>
</tr>
</tbody>
</table>

system with two zones shown in Figure 2. There are 2² = 4 tote classes. The release probabilities are set to ψ₁ = 0 and ψ₂ = 1/3, and the mean service times are μᵢ = 5 seconds for the entrance/exit and μᵢ = 15 seconds for the zones. Conveyor delays are deterministic and equal to μᵢ = 100 seconds. The number of order pickers dᵢ and dᵢ are equal to one, and the buffer sizes of the zones are q₁ = 1 and q₂ = 3, respectively.

Table 1 presents the average time in seconds a tote spends on the conveyor E(T_f(N)) = Σᵢ∈E E(T_i(n)) and in zones E(T_f(N)) = Σᵢ∈Z E(T_i(n)) and the overall throughput rate per hour X(N). These statistics are shown for the jump-over network (Jump), for the same closed queueing network with infinite buffers in the zones (CQN), and for the approximation of Yu and De Koster (2008) (YdK). YdK uses an open queueing network in its analysis. Using bisection, the rate at which this approximation is set such that the average number of totes in the open network is equal to N. The results show that the jump-over network produces more accurate results compared with the simulation of the original block-and-recirculate queueing network (Sim), in which the half width of the 95% confidence interval is given in brackets. In all cases, the algorithm stops after five iterations with ϵ = 10⁻³. Both CQN and YdK assume infinite buffers, which means that they cannot estimate the blocking probabilities. The run times for the Jump, CQN, and YdK methods are less than a second on a Core i7 with 2.4 GHz and 8 GB of RAM, whereas the simulation for N = 100 takes around 30 seconds (see Section 4 for more details on the simulation setup).

If the system contains a small number of totes N, the errors of the jump-over network are negligible and relatively small for CQN and YdK. This is obvious because almost no blocking occurs in the system; that is, only 5% of the totes that intend to visit the second zone are blocked. This means that the performance of CQN and YdK is similar to that of the jump-over network and the block-and-recirculate network. However, if the system contains a high number of totes, the blocking probability increases, and the totes have to recirculate more often.

If the number of totes in the system equals N = 40 or 50, blocking becomes more prominent. Because every zone is visited with the same frequency, more blocking occurs at zone 2 than at zone 1 because of differences in buffer sizes. Moreover, the system throughput time increases rapidly, whereas the throughput rate stabilizes because all the zones become saturated. CQN and YdK produce large errors in the average time a tote spends in the zones and at the conveyor nodes, which is due to the assumption of infinite buffers in the zones. This is not the case in the jump-over network. Because of recirculation, the conveyor nodes act as buffers for totes that cannot enter a zone. When N = 100, blocking negatively affects the performance of the system, and totes spend twice as long in the system compared with N = 50.

### 3. Multisegment Zone Picking Systems

This section presents the extension to multisegment zone-picking systems, which comprise three main parts: entrance/exit stations, conveyors, and zones. Figure 4 shows a zone picking system with four
segments and 11 zones. The workload control mechanism releases a tote, which travels to the first segment in which order lines have to be picked. The tote enters via the segment entrance station and stays in the segment until it has visited all required zones within the segment. When finished, the tote leaves the segment and is transported to another segment or to the system exit if the picking process has finished. The workload control mechanism controls the maximum number of totes in the system as well as the maximum number of totes within each segment. If a tote tries to enter a segment that is saturated, the control mechanism prevents it from entering. The blocked tote then skips the segment and stays on the main conveyor, potentially visiting other segments before again attempting to enter this segment. This is similar to zone blocking, but now blocking depends on the number of totes within an entire segment instead of in a single zone.

The multisegment zone-picking system is modeled as a closed queueing network with \( K \) segments. \( \mathcal{E} = \{ e_0, e_1, \ldots, e_k \} \) denotes the set of entrance/exit stations, where \( e_0 \) is the system entrance/exit and \( e_k \) is the entrance of segment \( k \), representing the conveyor connecting segment \( k \) with the main conveyor. Let \( \mathcal{Z} = \bigcup_{k=1}^{K} \mathcal{Z}^k \) is the set of zones, where \( \mathcal{Z}^k = \{ z_1^k, \ldots, z_{m_k}^k \} \) are the zones in segment \( k \). The total number of zones is \( M = \sum_{k=1}^{K} m_k \). The set of conveyor nodes is \( \mathcal{C} = \bigcup_{k=1}^{K} \mathcal{C}^k \), where \( \mathcal{C}^0 = \{ c^0_1, \ldots, c^0_{K+1} \} \) are the main conveyor nodes and \( \mathcal{C}^k = \{ c^k_1, \ldots, c^k_{m_k+1} \} \) are the conveyor nodes in segment \( k \). Finally, \( \mathcal{H} = \mathcal{C}^0 \cup \mathcal{C} \cup \mathcal{Z} \) is the set of all nodes in the network. Figure 5 shows the topology of the queueing network with \( K \) segments.

The system is partitioned into \( K + 1 \) subsystems: \( \{ \mathcal{H}^0, \mathcal{H}^1, \ldots, \mathcal{H}^k, \ldots, \mathcal{H}^K \} \), where \( \mathcal{H}^0 = \{ e_0 \} \cup \mathcal{C}^0 \) consists of the system entrance/exit and the nodes on the main conveyor, and \( \mathcal{H}^k = \{ e_k \} \cup \mathcal{C}^k \cup \mathcal{Z}^k \), the set of nodes belonging to the \( k \)th segment. We make the following assumptions:

- Each tote has a class \( r \subseteq \mathcal{Z} \) of zones to be visited and \( r^k \subseteq \mathcal{Z}^k \), \( k = 1, \ldots, K \), describes the zones a class \( r \) tote has to visit in segment \( k \). A class \( r \) tote enters segment \( k \) if and only if \( r^k \neq \emptyset \).
- Entrance station \( e_k \) to segment \( k \) is a delay node with an exponential delay with mean \( 1/\mu_k \), \( k = 1, \ldots, K \) that accounts for the time a tote needs for entering and leaving the segment.
- The maximum number of totes allowed in segment \( k \) is \( N_k \leq N_k, k = 1, \ldots, K \).

At system entrance \( e_0 \), new totes of class \( r \subseteq \mathcal{Z} \) are released with probability \( \psi_r \). After release, a class \( r \) tote is transported from \( e_0 \) to the first main conveyor node \( c^0_1 \). From \( c^0_0 \), the tote is either transported to

![Figure 4](https://example.com/figure4.png)
segment entrance \( e_k \) if \( r^k \neq \emptyset \) or moved to the next main conveyor node \( c^0_{k+1} \). If the number of totes in segment \( k \) equals \( N^k \), the tote skips segment \( k \) and also moves to \( c^0_{k+1} \) while its class remains the same. If the tote enters segment \( k \), it stays there until it has visited all required zones \( r^k \). When the tote leaves segment \( k \) via \( c^0_{k+1} \), its class changes from \( r \) upon entering segment \( k \) to \( s = r \setminus r^k \) upon leaving. After visiting the last main conveyor node \( c^0_k \), all totes with \( r \neq \emptyset \) are routed to \( c^0_1 \). Other totes are transported to exit \( e_0 \) and immediately replaced by a new one, waiting for release. The routing probabilities are formally described in Section EC.1.1.

The next step is to approximate the block-and-recirculate protocol by the jump-over blocking protocol in Section 3.1. The queueing network with jump-over blocking has a product-form stationary distribution, which is described in Section EC.1.2. The calculation of visit ratios is described in Section EC.1.3. Performance statistics of the jump-over network are calculated in Section 3.2 using flow-equivalent servers (Chandy et al. 1975) and MVA. An iterative algorithm to estimate the blocking probabilities is presented in Section 3.3.

3.1. Jump-Over Network

Totes can be blocked either by a zone or by a segment. Zone blocking has been described in Section 2. This section focuses on segment blocking, illustrated in Figure 6. In the block-and-recirculate protocol (Figure 6(a)), a class \( r \) tote that intends to visit segment \( k \), that is, \( r^k \neq \emptyset \), either enters if the number of totes in segment \( k \) is below \( N^k \) or skips segment \( k \). Class \( r \) totes skipping segment \( k \) maintain their class, and class \( r \) totes entering segment \( k \) always leave this segment as class \( r \setminus r^k \) totes.

In the jump-over blocking protocol (Figure 6(b)), a class \( r \) tote also skips segment \( k \) when it is full and proceeds as a class \( r \setminus r^k \) tote leaving segment \( k \).
A Bernoulli trial then determines whether each tote of class $r_1 r^k$ leaving segment $k$ changes class or not. It maintains class $r_1 r^k$ with probability $1 - B_k$ and reverts to class $r$ otherwise, independent of whether the tote visited or skipped segment $k$. However, totes leaving segment $k$ carry no information about which zones have been visited in segment $k$. To be able to revert to the correct class again, we add this information to the class description by including the initial class of the tote at the time of release from $e_0$. Hence, the routing probabilities of a class $\tilde{s} = \{h, s\}$ tote leaving segment $k$, that is, $s^i = 0$, are given by the following (cf. (9) and (10)):

$$
\begin{align*}
    p_{\tilde{s}^i, \tilde{s}^j, \tilde{s}^i, \tilde{s}^j | \tilde{s}} &= 1 - B_k, & k = 1, \ldots, K, \tilde{s} = \{h, s\}, s^i = 0, \\
    p_{\tilde{s}^i, \tilde{s}^j, \tilde{s}^i, \tilde{s}^j | \tilde{s}} &= B_k, & k = 1, \ldots, K, \tilde{s} = \{h, s\}, s^i = 0,
\end{align*}
$$

where $\tilde{r} = \{h, s \cup h^k\}$.

The flows of class $r$ and class $r_1 r^k$ totes entering $c_{k+1}^0$ match under both protocols by choosing $B_k$ as the fraction of totes skipping segment $k$ in the block-and-recirculate network. In other words, $B_k$ is the blocking probability of segment $k$ under block and recirculate. It is estimated in Section 3.3.

### 3.2. Aggregation Technique

We can analyze the jump-over network by directly applying MVA as described in Section 2.4. However, it is computationally more efficient to first apply Norton’s theorem by replacing all segments by flow-equivalent server centers with load-dependent exponential service rates (Chandy et al. 1975, Walrand 1983, Boucherie 1998) and then applying MVA. Norton’s equivalent of the jump-over network is constructed by replacing each segment $k$ by a flow equivalent server with exponential service rate

$$
\mu_{FES_k}(n) = X^k(n), \quad n = 1, \ldots, N^k, k = 1, \ldots, K,
$$

where $X^k(n)$ is the throughput of segment $k$ in isolation when it contains $n$ totes. Segment $k$ is isolated by short-circuiting all nodes outside segment $k$. This means that a tote leaving segment $k$ through conveyor node $c_{n+1}^k$ is directly routed back to entrance $e_k$. The throughput of this isolated segment can be evaluated by MVA (where the entrance is now a delay node).

Figure 7 shows Norton’s equivalent of Figure 5, which is identical to Figure 2(b) except that the zones are replaced by flow-equivalent servers. Norton’s network can be analyzed by MVA, where Equation (46) should be replaced by the mean throughput time in the $k$th flow-equivalent server (Reiser 1981),

$$
E(T_{FES_k}(n)) = \sum_{j=1}^{N_k} \frac{j}{\mu_{FES_k}(j)} \pi_{FES_k}(j-1|n-1), \quad k = 1, \ldots, K,
$$

where $\pi_{FES_k}(j|n)$ are the marginal queue length probabilities of having $j$ totes in the $k$th flow-equivalent server in a network with $n$ circulating totes. These probabilities can be obtained through a balance argument, similar to in (49),

$$
\pi_{FES_k}(j|n) = \frac{V_{FES_k}(X(n))}{\mu_{FES_k}(j)} \pi_{FES_k}(j-1|n-1),
$$

where $V_{FES_k}$ is the visit ratio of the $k$th flow-equivalent server, which is equal to the visit ratio of entrance $e_k$. Equation (55) is obtained by applying Little’s law and substitution of (56).

Performance statistics in Norton’s network correspond to aggregated performance statistics of the segments, for example, $E(T_{FES_k}(n))$ is the mean throughput time of segment $k$, and $\pi_{FES_k}(j|N)$ are the marginal probabilities of having $j$ totes in segment $k$ when there are $N$ totes in the jump-over network. Detailed performance statistics of nodes within segments can be retrieved by disaggregation. Let $\pi_i(j|n)$ be the probability of $j$ totes in node $i \in \mathcal{H}^k$ given there are $n$ totes in segment $k$. The detailed queue length probabilities $\pi_i(j|N)$ are then given by (Baynat and Dallery 1993):

$$
\pi_i(j|N) = \begin{cases} 
\Pi_i(j|N), & \text{if } i \in \mathcal{H}^0, \\
\sum_{l=1}^{N^k} \pi_l(j|l) \Pi_{FES_k}(l|N), & \text{if } i \in \mathcal{H}^k.
\end{cases}
$$

**Figure 7.** The Norton Equivalent of the Jump-Over Network

![Diagram of Norton Equivalent](image)

Note. Segments are replaced by a flow-equivalent server with load-dependent exponential service rates.
Performance statistics of all nodes can now be calculated as follows. Utilization of system entrance \( e_0 \) (with \( d_0 = 1 \)) and zones \( i \in \mathcal{I} \) is given by

\[
p_i = 1 - \frac{d_i}{d_1} - j \pi_i(j|N), \quad i \in e_0 \cup \mathcal{I}.
\]

(58)

The mean number of totes in node \( i \in \mathcal{I} \) is given by

\[
E(L_i(N)) = \frac{\sum_{j=1}^{d_i} \pi_i(j|N)}{i \in \mathcal{I},}
\]

(59)

where \( \sigma_i = N \) if \( i \in \mathcal{I}_0 \), and \( \sigma_i = N^k \) if \( i \in \mathcal{I}^k \backslash \mathcal{I}^k \) and \( d_i + q_i \) if \( i \in \mathcal{I}^k \). Applying Little’s law yields

\[
E(T_i(N)) = E(L_i(N))/V/X(N), \quad i \in \mathcal{I},
\]

(60)

where \( X(N) \) is the throughput rate of Norton’s network.

### 3.3. Iterative Algorithm for Calculating the Blocking Probabilities

In the jump-over network, totes leaving segment \( k \) or zone \( i \) randomly change class according to probability \( B_k \) and \( b_i \), respectively. The idea is to match \( B_k \) and \( b_i \) with the blocking probability of segment \( k \) and zone \( i \) in the block-and-recirculate network, which can be iteratively estimated as follows. Initially, \( B_k^{(1)} = 0, \ k = 1, \ldots, K \), and \( b_i^{(1)} = 0, \ i \in \mathcal{I} \). Then we calculate the marginal queue length probabilities using Equation (56) in Norton’s network and estimate \( B_k \) as a fraction of arrivals finding \( N^k \) totes in flow-equivalent server \( k \). By the arrival theorem,

\[
B_k^{(m+1)} = \Pi_{k}^{(m)}(N^k|N-1), \quad k = 1, \ldots, K,
\]

(61)

where the superscript \( (m) \) corresponds to the iteration. Accordingly, by using the detailed marginal queue length probabilities of Equation (57), we get

\[
b_i^{(m+1)} = \pi_i^{(m)}(d_i + q_i|N^k - 1, N - 1),
\]

\[
= \sum_{l=1}^{N^k - 1} \pi_i(l|d_i + q_i | k) \Pi_{k}^{(m)}(l|N-1), \quad i \in \mathcal{I},
\]

(62)

which is the probability of finding a full buffer in zone \( i \) in a network with \( N - 1 \) totes, where at most \( N^k - 1 \) totes are allowed in segment \( k \). Note that we apply a modification of the “usual” arrival theorem: an arriving tote in zone \( i \) not only sees the network in equilibrium without itself but also one in which the capacity of segment \( k \) containing zone \( i \) is reduced by one.

Subsequently, routing probabilities and visit ratios are recalculated and so on. This procedure is repeated until the differences \( B_k^{(m+1)} - B_k^{(m)} \) and \( b_i^{(m+1)} - b_i^{(m)} \) for all \( k \) and \( i \) are less than some \( \epsilon \).

### 4. Numerical Results

We compare the results of the jump-over approximation with a discrete-event simulation of the block-and-recirculate network. This section is split into two parts. Section 4.1 is devoted to single-segment routing systems and Section 4.2 to multisegment routing systems. Both the jump-over network and the discrete-event simulation are implemented in Java. In each case, the simulation is run 10 times for 1,000,000 seconds, preceded by 10,000 seconds of initialization for the system to become stable. This guarantees that the width of the 95% confidence interval of the system throughput time is less than 1% of the mean value. All experiments are run on Core i7 with 2.4 GHz and 8 GB of RAM.

#### 4.1. Single-Segment Systems

This section explores the performance of the approximation for the single-segment system. Table 2 lists the parameters of the test set. The number of zones \( M \) varies between one and eight, and the number of totes \( N \) varies between 10 and 80. We first assume that all zones and conveyor nodes are identical and that all tote classes are released into the system with equal probability. This ensures that the workload of all zones is balanced. In the test set, conveyor delays are deterministic and \( \mu_i^{-1}, i \in \mathcal{C} \) varies between 20 and 60 seconds. Picking times are exponential, and \( \mu_i^{-1}, i \in \mathcal{I} \) varies between 10 and 30 seconds. The number of order pickers \( d_i, i \in \mathcal{I} \) and buffer places \( q_i, i \in \mathcal{I} \) varies between one and three and zero.

**Table 2. Parameters of the Test Set for the Single-Segment System**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of zones, ( M )</td>
<td>1, 2, 3, 4, 5, 6, 7, 8</td>
</tr>
<tr>
<td>Number of totes, ( N )</td>
<td>10, 20, 30, …, 80</td>
</tr>
<tr>
<td>Conveyor delays, ( \mu_i^{-1}, i \in \mathcal{C} )</td>
<td>20, 30, 40, 50, 60</td>
</tr>
<tr>
<td>Mean picking times, ( \mu_i^{-1}, i \in \mathcal{I} )</td>
<td>10, 15, 20, 25, 30</td>
</tr>
<tr>
<td>Number of order pickers, ( d_i, i \in \mathcal{I} )</td>
<td>1, 2, 3</td>
</tr>
<tr>
<td>Buffer size of a zone, ( q_i, i \in \mathcal{I} )</td>
<td>0, 1</td>
</tr>
</tbody>
</table>

(a) Balanced test set (9,600 cases)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of zones, ( M )</td>
<td>2, 3, 4, 5, 6, 7, 8</td>
</tr>
<tr>
<td>Number of totes, ( N )</td>
<td>10, 20, 30, …, 80</td>
</tr>
<tr>
<td>Mean picking times, ( \mu_i^{-1}, i \in \mathcal{I} )</td>
<td>• 10, 10, 10, …</td>
</tr>
<tr>
<td>• 10, 12, 14, …</td>
<td></td>
</tr>
<tr>
<td>• 10, 15, 20, …</td>
<td></td>
</tr>
<tr>
<td>• 10, 20, 30, …</td>
<td></td>
</tr>
</tbody>
</table>

(b) Imbalanced test set (224 cases)
and one, respectively. In total, this leads to 9,600 cases.

We also examine the effect of workload imbalance resulting from differences in mean picking times among zones. In this test set, the deterministic delays $\mu^{-1}, i \in I$ are equal to 30 seconds, and both the number of order pickers $d_i, i \in I$ and buffer places $q_i, i \in I$ are equal to one. Four different scenarios are created for the mean picking times. In the first scenario, the mean picking times are equal, whereas in the other three scenarios, they increase by either 2, 5, or 10 seconds per subsequent zone. This leads to an additional 224 cases. The run time per case for the analytical model is less than a second, whereas simulation takes at most 30 seconds for the larger systems.

The results of the balanced test set are summarized in Tables 3–6. Each table lists the average relative error between approximation and simulation of system throughput in hour$^{-1}$, the average number of circulations of a tote before moving to the exit, and the sum of mean throughput times of the zones. Almost all errors are between 0–1%, with only a few larger than 5%.

Tables 3 and 4 show that the largest errors occur when the system has three or four zones and when the number of totes in the system is high. An explanation is that the probability of blockages increases with a lower number of zones $M$ or with a higher number of totes $N$ in the system. Moreover, if blocking is prevalent, more zones implies that the approximation needs to estimate more blocking probabilities. This creates more room for error. Eventually, $M$ is high enough for blocking to be almost fully absent for any $N$. The approximation becomes exact because the network behaves precisely as the block-and-recirculate network, in which totes are hardly ever blocked.

Tables 5 and 6 show that the largest errors occur with low conveyor delays and high mean picking times. Here the product-form assumption that each node can be analyzed in isolation does not describe the real behavior adequately. For example, if a tote is blocked by a zone, it can circulate through the entire system and eventually encounter the zone still working on the same tote. This creates dependencies between successive visits to the nodes that are not captured by the approximation. However, this situation is unlikely in practice. The total recirculation time is usually much higher than the time a tote spends in a zone.

TABLE 3. Results of the Balanced Test Set with a Varying Number of Zones $M$

<table>
<thead>
<tr>
<th>$M$</th>
<th>Average</th>
<th>0–1</th>
<th>1–5</th>
<th>&gt; 5</th>
<th>Average</th>
<th>0–1</th>
<th>1–5</th>
<th>&gt; 5</th>
<th>Average</th>
<th>0–1</th>
<th>1–5</th>
<th>&gt; 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.08</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.08</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.09</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.67</td>
<td>70.0</td>
<td>29.8</td>
<td>0.2</td>
<td>0.78</td>
<td>69.0</td>
<td>29.8</td>
<td>1.3</td>
<td>0.44</td>
<td>83.9</td>
<td>16.1</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>0.78</td>
<td>68.2</td>
<td>31.7</td>
<td>0.2</td>
<td>0.94</td>
<td>67.2</td>
<td>30.3</td>
<td>2.5</td>
<td>0.44</td>
<td>86.2</td>
<td>13.8</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>0.73</td>
<td>71.9</td>
<td>27.8</td>
<td>0.3</td>
<td>0.90</td>
<td>71.3</td>
<td>25.9</td>
<td>2.8</td>
<td>0.38</td>
<td>90.3</td>
<td>9.8</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>0.64</td>
<td>76.6</td>
<td>23.3</td>
<td>0.2</td>
<td>0.80</td>
<td>75.0</td>
<td>22.4</td>
<td>2.6</td>
<td>0.32</td>
<td>93.2</td>
<td>6.8</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>0.54</td>
<td>80.4</td>
<td>19.5</td>
<td>0.1</td>
<td>0.68</td>
<td>78.6</td>
<td>18.9</td>
<td>2.5</td>
<td>0.28</td>
<td>94.9</td>
<td>5.1</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>0.45</td>
<td>83.8</td>
<td>16.2</td>
<td>0.0</td>
<td>0.57</td>
<td>82.4</td>
<td>15.8</td>
<td>1.8</td>
<td>0.25</td>
<td>96.9</td>
<td>3.1</td>
<td>0.0</td>
</tr>
<tr>
<td>8</td>
<td>0.38</td>
<td>86.7</td>
<td>13.3</td>
<td>0.0</td>
<td>0.48</td>
<td>85.2</td>
<td>13.5</td>
<td>1.3</td>
<td>0.23</td>
<td>97.7</td>
<td>2.3</td>
<td>0.0</td>
</tr>
</tbody>
</table>

TABLE 4. Results of the Balanced Test Set with a Varying Number of Totes in the System $N$

<table>
<thead>
<tr>
<th>$N$</th>
<th>Average</th>
<th>0–1</th>
<th>1–5</th>
<th>&gt; 5</th>
<th>Average</th>
<th>0–1</th>
<th>1–5</th>
<th>&gt; 5</th>
<th>Average</th>
<th>0–1</th>
<th>1–5</th>
<th>&gt; 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.24</td>
<td>95.0</td>
<td>4.8</td>
<td>0.2</td>
<td>0.29</td>
<td>93.1</td>
<td>5.9</td>
<td>1.0</td>
<td>0.21</td>
<td>99.8</td>
<td>0.3</td>
<td>0.0</td>
</tr>
<tr>
<td>20</td>
<td>0.40</td>
<td>86.8</td>
<td>12.9</td>
<td>0.3</td>
<td>0.53</td>
<td>85.5</td>
<td>12.3</td>
<td>2.2</td>
<td>0.21</td>
<td>97.9</td>
<td>2.1</td>
<td>0.0</td>
</tr>
<tr>
<td>30</td>
<td>0.52</td>
<td>81.7</td>
<td>18.1</td>
<td>0.3</td>
<td>0.67</td>
<td>80.0</td>
<td>17.5</td>
<td>2.5</td>
<td>0.24</td>
<td>95.6</td>
<td>4.4</td>
<td>0.0</td>
</tr>
<tr>
<td>40</td>
<td>0.59</td>
<td>77.6</td>
<td>22.3</td>
<td>0.2</td>
<td>0.74</td>
<td>76.7</td>
<td>20.8</td>
<td>2.6</td>
<td>0.28</td>
<td>93.0</td>
<td>7.0</td>
<td>0.0</td>
</tr>
<tr>
<td>50</td>
<td>0.62</td>
<td>75.3</td>
<td>24.8</td>
<td>0.0</td>
<td>0.77</td>
<td>74.2</td>
<td>23.7</td>
<td>2.2</td>
<td>0.32</td>
<td>91.8</td>
<td>8.3</td>
<td>0.0</td>
</tr>
<tr>
<td>60</td>
<td>0.64</td>
<td>74.1</td>
<td>25.9</td>
<td>0.0</td>
<td>0.76</td>
<td>73.0</td>
<td>25.1</td>
<td>1.9</td>
<td>0.36</td>
<td>89.6</td>
<td>10.4</td>
<td>0.0</td>
</tr>
<tr>
<td>70</td>
<td>0.64</td>
<td>73.6</td>
<td>26.4</td>
<td>0.0</td>
<td>0.75</td>
<td>72.9</td>
<td>25.7</td>
<td>1.4</td>
<td>0.38</td>
<td>88.8</td>
<td>11.3</td>
<td>0.0</td>
</tr>
<tr>
<td>80</td>
<td>0.64</td>
<td>73.6</td>
<td>26.4</td>
<td>0.0</td>
<td>0.72</td>
<td>73.3</td>
<td>25.7</td>
<td>1.0</td>
<td>0.41</td>
<td>86.7</td>
<td>13.3</td>
<td>0.0</td>
</tr>
</tbody>
</table>
because the probability of blockages is higher, which increases errors, as seen in the previous tables. Still, on average, the errors for the three statistics are well below 1%.

4.2. Multisegment Systems

We create a new test set for the multisegment system. Table 8 lists its parameters. In all test cases, the number of zones $M$ equals 18, and the number of zones per segment $m_k$ varies between three, six, and nine. Zones and conveyor nodes are identical within every segment, that is, $\mu^{-1}_i = 30, i \in \mathcal{E}\setminus\mathcal{E}^0, \mu^{-1}_i = 15, i \in \mathcal{E}$, and $q_i = d_i = 1, i \in \mathcal{E}$. Release probabilities $\psi_r$ are the same for all $r$, and the service means of all entrances are equal to $\mu^{-1}_i = 5, i \in \mathcal{E}$. Picking times and system entrance times are exponential. Delays in conveyor nodes and segment entrances are deterministic. The number of totes in the system varies between 10 and 80, and the capacities of the segments $N^k$ vary between 10 and 40 totes as long as $N \geq N^k$. The main conveyor times, $\mu^{-1}_i, i \in \mathcal{E}^0$ vary between 10 and 60. This leads to 1,320 test cases. The run time per case for the analytical model is around 10 seconds, whereas simulation takes at most one minute for larger systems.

The results of the multisegment test set are summarized in Tables 9 and 10. The overall average error is 0.21% for system throughput, 0.93% for the mean number of circulations on the main conveyor, and 0.24% for the average throughput times of the zones. The tables show that the errors are the largest in cases with a low number of segments and a fast main conveyor. In these cases, totes are more likely to be blocked by a segment and subsequently need to recirculate on the main conveyor multiple times. As seen in the previous results, errors increase when there are more blockages in the system. Similar results can be seen when varying the segment capacities $N^k$.

Comparing the systems shows that, if the main conveyors are slow ($\mu^{-1}_i \geq 50$ seconds) and if the system contains a low number of totes ($N \leq 20$), the systems with $m^k = 9, k = 1, 2$ obtain the highest throughput because of less delay on the conveyors. However, if the segment capacity $N^k \leq 20$ is low, $m^k = 3, k = 1, \ldots, 6$ has the highest throughput because the probability that a tote is blocked by a segment is lower, and less recirculation of totes is required.

A further validation for a real-life system with nonexponential picking times and multiple segments can be found in Section EC.2.

5. Conclusion and Further Research

In this paper, we study sequential zone picking systems with single-segment routing and those with multisegment routing. We propose a queueing network to estimate the throughput capacity. Because an exact analysis of this queueing network is not feasible, we approximate its blocking protocol with the jump-over protocol. This network admits a product-form solution. We use MVA and an aggregation technique to obtain accurate performance estimates. Results

### Table 5. Results of the Balanced Test Set with Varying Conveyor Delays $\mu^{-1}_i, i \in \mathcal{E}$

<table>
<thead>
<tr>
<th>$\mu^{-1}_i$</th>
<th>Error % in system throughput</th>
<th>Error % in number of circulations</th>
<th>Error % in throughput times zones</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average 0–1 1–5 &gt; 5</td>
<td>Average 0–1 1–5 &gt; 5</td>
<td>Average 0–1 1–5 &gt; 5</td>
</tr>
<tr>
<td>20</td>
<td>0.91 66.5 33.0 0.5</td>
<td>1.21 65.3 28.1 6.7</td>
<td>0.47 85.4 14.6 0.0</td>
</tr>
<tr>
<td>30</td>
<td>0.64 74.1 25.9 0.0</td>
<td>0.79 72.7 25.2 2.2</td>
<td>0.33 91.0 9.0 0.0</td>
</tr>
<tr>
<td>40</td>
<td>0.47 81.4 18.6 0.0</td>
<td>0.55 80.3 19.4 0.4</td>
<td>0.27 94.4 5.6 0.0</td>
</tr>
<tr>
<td>50</td>
<td>0.36 86.3 13.7 0.0</td>
<td>0.41 85.5 14.5 0.0</td>
<td>0.23 96.3 3.8 0.0</td>
</tr>
<tr>
<td>60</td>
<td>0.29 90.2 9.8 0.0</td>
<td>0.31 89.2 10.8 0.0</td>
<td>0.21 97.3 2.7 0.0</td>
</tr>
</tbody>
</table>

### Table 6. Results of the Balanced Test Set with Varying Mean Picking Times $\mu^{-1}_i, i \in \mathcal{E}$

<table>
<thead>
<tr>
<th>$\mu^{-1}_i$</th>
<th>Error % in system throughput</th>
<th>Error % in number of circulations</th>
<th>Error % in throughput times zones</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average 0–1 1–5 &gt; 5</td>
<td>Average 0–1 1–5 &gt; 5</td>
<td>Average 0–1 1–5 &gt; 5</td>
</tr>
<tr>
<td>10</td>
<td>0.24 93.0 7.0 0.0</td>
<td>0.23 92.7 7.3 0.0</td>
<td>0.17 98.2 1.8 0.0</td>
</tr>
<tr>
<td>15</td>
<td>0.40 84.9 15.1 0.0</td>
<td>0.46 83.5 16.1 0.4</td>
<td>0.23 95.5 4.5 0.0</td>
</tr>
<tr>
<td>20</td>
<td>0.55 78.4 21.6 0.0</td>
<td>0.67 76.8 21.7 1.5</td>
<td>0.30 92.7 7.3 0.0</td>
</tr>
<tr>
<td>25</td>
<td>0.68 72.9 27.0 0.2</td>
<td>0.86 71.8 25.3 2.9</td>
<td>0.37 90.1 9.9 0.0</td>
</tr>
<tr>
<td>30</td>
<td>0.80 69.3 30.3 0.4</td>
<td>1.05 68.1 27.3 4.5</td>
<td>0.43 87.8 12.2 0.0</td>
</tr>
</tbody>
</table>
indicate that the relative error in the throughput estimate is typically less than 1% compared with simulation.

We suggest some directions for further research.

- Operational policies. The model can be used to evaluate and compare the throughput capacity of operational policies, such as order batching and order splitting, as in Yu and De Koster (2008). An optimization framework can be formulated for the allocation of products to zones to maximize throughput capacity.
- Flexible order pickers. The model can be extended to study the impact of flexible order pickers in adjacent zones, helping each other when the workload in one of the zones is temporarily high.
- Merging of totes. When the zone picking system is heavily loaded, congestion can occur when streams of totes merge. For example, totes leaving a zone

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of segments, $K$</td>
<td>2, 3, 4, 5, 6</td>
</tr>
<tr>
<td>Number of totes, $N$</td>
<td>10, 20, ..., 80</td>
</tr>
<tr>
<td>Mean main conveyor times, $\mu_i^{-1}, i \in \mathbb{E}$</td>
<td>10, 20, 30, 40, 50, 60</td>
</tr>
<tr>
<td>Segment capacity, $N_k, k = 1, \ldots, K$</td>
<td>10, 15, 20, 25, 30, 35, 40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of zones per segment, $m_k$</td>
<td><em>9, 9</em></td>
</tr>
<tr>
<td><em>6, 6, 6</em></td>
<td></td>
</tr>
<tr>
<td><em>3, 6, 3, 6</em></td>
<td></td>
</tr>
<tr>
<td><em>3, 3, 6, 3, 3</em></td>
<td></td>
</tr>
</tbody>
</table>

Table 7. Results of the Imbalanced Test Set with Varying Mean Picking Times $\mu_i^{-1}, i \in \mathbb{E}$

<table>
<thead>
<tr>
<th>$\mu_i^{-1}$</th>
<th>Error % in system throughput</th>
<th>Error % in number of circulations</th>
<th>Error % in throughput times zones</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average 0–1 1–5 &gt; 5</td>
<td>Average 0–1 1–5 &gt; 5</td>
<td>Average 0–1 1–5 &gt; 5</td>
</tr>
<tr>
<td>10, 10, 10, ...</td>
<td>0.20 100.0 0.0 0.0</td>
<td>0.20 100.0 0.0 0.0</td>
<td>0.15 100.0 0.0 0.0</td>
</tr>
<tr>
<td>10, 12, 14, ...</td>
<td>0.23 100.0 0.0 0.0</td>
<td>0.24 100.0 0.0 0.0</td>
<td>0.16 100.0 0.0 0.0</td>
</tr>
<tr>
<td>10, 15, 20, ...</td>
<td>0.35 98.2 1.8 0.0</td>
<td>0.36 94.6 5.4 0.0</td>
<td>0.21 100.0 0.0 0.0</td>
</tr>
<tr>
<td>10, 20, 30, ...</td>
<td>0.40 89.3 10.7 0.0</td>
<td>0.45 85.7 14.3 0.0</td>
<td>0.32 100.0 0.0 0.0</td>
</tr>
</tbody>
</table>

Table 8. Parameters of the Multisegment Routing Model Test Set (1,320 Cases)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
</table>
| Table 9. Results of the Multisegment Routing Test Set with a Varying Number of Zones per Segment $m_k$

<table>
<thead>
<tr>
<th>$m_k$</th>
<th>Error % in system throughput</th>
<th>Error % in number of circulations</th>
<th>Error % in throughput times zones</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average 0–1 1–5 &gt; 5</td>
<td>Average 0–1 1–5 &gt; 5</td>
<td>Average 0–1 1–5 &gt; 5</td>
</tr>
<tr>
<td>9, 9</td>
<td>0.21 99.6 0.4 0.0</td>
<td>2.27 66.3 20.1 13.6</td>
<td>0.23 100.0 0.0 0.0</td>
</tr>
<tr>
<td>6, 6, 6</td>
<td>0.25 100.0 0.0 0.0</td>
<td>1.23 72.0 21.6 6.4</td>
<td>0.23 100.0 0.0 0.0</td>
</tr>
<tr>
<td>6, 3, 6, 3</td>
<td>0.19 100.0 0.0 0.0</td>
<td>0.43 84.8 15.2 0.0</td>
<td>0.24 100.0 0.0 0.0</td>
</tr>
<tr>
<td>3, 3, 6, 3, 3</td>
<td>0.16 100.0 0.0 0.0</td>
<td>0.28 89.8 10.2 0.0</td>
<td>0.25 100.0 0.0 0.0</td>
</tr>
<tr>
<td>3, 3, 3, 3, 3, 3</td>
<td>0.23 99.2 0.8 0.0</td>
<td>0.43 86.7 12.5 0.8</td>
<td>0.23 100.0 0.0 0.0</td>
</tr>
</tbody>
</table>

Table 10. Results of the Multisegment Routing Test Set with Varying Mean Conveyor Times $\mu_i^{-1}, i \in \mathbb{E}$

<table>
<thead>
<tr>
<th>$\mu_i^{-1}, i \in \mathbb{E}$</th>
<th>Error % in system throughput</th>
<th>Error % in number of circulations</th>
<th>Error % in throughput times zones</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average 0–1 1–5 &gt; 5</td>
<td>Average 0–1 1–5 &gt; 5</td>
<td>Average 0–1 1–5 &gt; 5</td>
</tr>
<tr>
<td>10</td>
<td>0.28 99.1 0.9 0.0</td>
<td>2.23 66.8 22.3 10.9</td>
<td>0.23 100.0 0.0 0.0</td>
</tr>
<tr>
<td>20</td>
<td>0.24 100.0 0.0 0.0</td>
<td>1.15 75.5 18.6 5.9</td>
<td>0.23 100.0 0.0 0.0</td>
</tr>
<tr>
<td>30</td>
<td>0.21 100.0 0.0 0.0</td>
<td>0.77 77.3 18.2 4.5</td>
<td>0.24 100.0 0.0 0.0</td>
</tr>
<tr>
<td>40</td>
<td>0.19 99.5 0.5 0.0</td>
<td>0.58 83.6 14.5 1.8</td>
<td>0.24 100.0 0.0 0.0</td>
</tr>
<tr>
<td>50</td>
<td>0.18 100.0 0.0 0.0</td>
<td>0.45 86.8 11.8 1.4</td>
<td>0.24 100.0 0.0 0.0</td>
</tr>
<tr>
<td>60</td>
<td>0.16 100.0 0.0 0.0</td>
<td>0.37 89.5 10.0 0.5</td>
<td>0.24 100.0 0.0 0.0</td>
</tr>
</tbody>
</table>
have to wait for a sufficiently large open space on the conveyor before merging. This waiting time can be viewed as blocking after service and can be incorporated into the model to assess its impact on the throughput capacity (van der Gaast et al. 2018).

- External order arrivals. To estimate the throughput capacity, we assume that when a tote has collected all required items for a customer order, a new order tote can be released immediately; that is, there are always customer orders available. In an operational setting, the arrival rate $\lambda$ of customer orders is lower than the throughput capacity, so it may happen that a new order is not available upon completion of a tote. Hence, the number of circulating totes is not always $N$, but can also be less. An appropriate model to study this situation is a semiopen queueing network (see Jia and Heragu (2009)), for which the proposed closed queueing network may be used as a building block as follows. For each population $n = 0, \ldots, N$, the MVA algorithm in Sections 2.4 and 3.2 provides the throughput $X(n)$. Then the semiopen queueing network is replaced by a flow-equivalent server with service rate $\mu_n = X(n)$ for $n < N$ and $\mu_n = X(N)$ for $n \geq N$, and the resulting birth-and-death process with birth rate $\lambda$ and death rate $\mu_n$, can be solved to obtain relevant performance measures, such as mean order flow times.

Our approach to model and analyze queueing networks with blocking may be applied in many applications beyond zone picking systems, for example, end-of-aisle picking systems, AGV transportation systems, and vehicle-based compact storage systems.

References


Jelmer P. van der Gaast is an assistant professor in the Department of Management Science at Fudan University, Shanghai. His research interests include warehouse operations, stochastic processes, optimization, and queueing theory.

René B. M. de Koster is a professor of logistics and operations management at Rotterdam School of Management, Erasmus University. His research interests are warehousing, terminal, and behavioral operations.

Ivo J. B. F. Adan is a full professor in the sections operations, planning, accounting and control (Department of Industrial Engineering and Innovation Sciences) and dynamics and control (Department of Mechanical Engineering) at Eindhoven University of Technology and holds the Manufacturing Networks Chair. His expertise and tuition areas include probability theory/statistics, operations research, manufacturing networks, stochastic operations research, and queueing models.

Jacques A. C. Resing is an assistant professor in the stochastic operations research group at Eindhoven University of Technology. His research is, in general, focused on applied probability and, in particular, on queueing theory. Among others, he studies several polling systems, fluid queues, and tandem queues.