

Vertical or Horizontal Transport? - Comparison of robotic storage and retrieval systems

Kaveh Azadeh^{1*}, Debjit Roy², René de Koster¹

¹Rotterdam School of Management, Erasmus University, The Netherlands

²Indian Institute of Management Ahmedabad, India

*E-mail: azadeh@rsm.nl

Abstract

Autonomous vehicle-based storage and retrieval systems are commonly used in e-commerce fulfillment as they allow a high and flexible throughput capacity. In these systems, roaming robots transport loads between a storage location and a workstation. Two main variants exist: *Horizontal*, where the robots only move horizontally and use lifts for vertical transport and a new variant *Vertical*, where the robots can also travel vertically in the rack. This paper builds a framework to analyze the performance of the vertical system and to compare its throughput capacity with the horizontal system. We build closed-queueing network models for this that in turn are used to optimize the design. The results show that the optimal height-to-width ratio of a vertical system is around 1. As a large number of system robots may lead to blocking and delays, we compare the effect of two different robot blocking protocols on the system throughput: robot Recirculation and Wait-On-Spot. The Wait-On-Spot policy produces a higher system throughput when the number of robots in the system is small. However, for a large number of robots in the system, the Recirculation policy dominates the Wait-On-Spot policy. Finally, we compare the operational costs of the vertical and the horizontal transport system. For systems with one load/unload (L/U) point, the vertical system always produces a similar or higher system throughput, with a lower operating cost compared to the horizontal system with a discrete lift. It also outperforms the horizontal system with a continuous lift in systems with two L/U points.

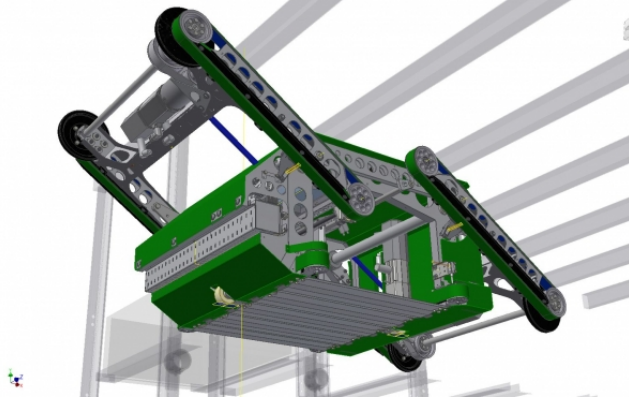
Key words: Internal transportation, robotic technology, queueing networks, performance analysis, blocking

1 Introduction

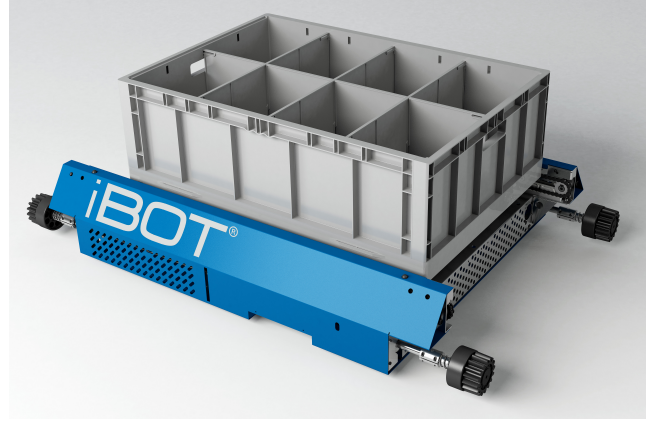
The main challenge in e-commerce fulfillment is to adapt the picking capacity to the order volume required. The Shuttle- or Autonomous Vehicle-based Storage and Retrieval System (AVS/RS) is one candidate to address this challenge. In this system, the combination of autonomous shuttles and lifts are used to perform the order fulfillment process. Typically, the system throughput is constrained by the number of lifts in this system. Shuttles move in the horizontal direction using rails and are transported in the vertical direction using lifts. Hence, we categorize them as *Horizontal* systems.

Recently, robotic-based storage and retrieval systems have been developed that eliminate the multi-touch retrieval process of AVS/R systems. In these systems, a single robot can independently roam throughout the storage rack to transport items between storage locations and workstations. Two variants of robots used in these systems. In the first group, robots move independently in horizontal and diagonal directions to access a storage location, which we categorize as *Diagonal* systems. One example of the diagonal system is a climbing robot called Rack RacerTM(see Figure 1a) developed by Fraunhofer IML. In the second group, which is the focus of this study, robots move independently in the horizontal and vertical directions inside the rack structure. Therefore we categorize them as *Vertical* systems. Perfect PickTM developed by OPEX Corporations is one example of a vertical system, using the iBotTM(see Figure 1b) as a robot. Since its inception in 2013, vertical systems have been installed at several e-retailer warehouses in the U.S., such as iHerb, BHFO, NewEgg, and Petzl.

The single-touch retrieval process gives the vertical system an edge over its horizontal counterpart when it comes to flexibility and throughput adjustment. In the vertical system, the desired throughput level can be obtained by only choosing the correct number of robots. However, in the horizontal system, the number of shuttles, as well as the number of lifts, need to be adjusted to achieve a certain throughput. Furthermore, adding additional lifts requires a major overhaul of the system. Moreover, if one of the robots breaks down in the vertical system, it can be replaced without affecting the operation. In contrast, a failure of an exchange point in the horizontal sys-



(a) RackRacerTM(Fraunhofer 2014)



(b) iBotTM(OPEX 2013)

Figure 1: Robots in single-touch systems

tem could result in a system shutdown. Although it seems that the vertical system is more flexible and reliable compared to the horizontal system, it is not clear which system achieves a higher throughput performance with the same storage capacity and number of robots. As a result, a comprehensive study on both systems is required to make that judgment. Many studies exist that analyze the performance of the horizontal systems (e.g. Malmborg (2002), Marchet et al. (2012), and Roy et al. (2012)), while the vertical system has not been studied yet. Therefore, we first need to investigate the vertical system before we can compare the performance and costs of the two systems.

The throughput capacity of the vertical system depends on several design choices, in particular, the layout configuration and the number of operating robots. Although increasing the number of robots increases the system throughput capacity, it can simultaneously lead to increased blocking delays, potentially reducing the throughput capacity. Therefore, an understanding of the effect of blocking and mitigating policies on performance is crucial, especially in the conceptual design phase. To evaluate the different blocking policies and system designs, one obvious approach is to build detailed simulation models. However, developing a realistic and detailed simulation model for analyzing all possible design scenarios and parameter settings is very time consuming. Therefore, at the early conceptualization stage, analytical models are used to reduce the design search space and identify a few promising configurations, which can then be fine-tuned using simulation. These analytical models are faster to evaluate, and they allow optimization by enumeration over a large

number of design parameters (Tappia et al. 2017, Zaerpour et al. 2015).

Hence, the objective of this paper is to answer the following research questions:

1. How to build accurate and efficient analytical models to analyze the performance of the vertical system?
2. What is an optimal layout for the vertical system in terms of throughput performance?
3. How do the blocking delays affect the throughput performance of the vertical system?
4. Which system is better in terms of costs and throughput capacity: horizontal or vertical?

The focus and prime contribution of this paper lies in the model formulation, analysis, and system comparison. The prime performance metric during the conceptualization phase is the system throughput. We build a closed-queueing network to model and use the system and Approximate Mean Value Analysis (AMVA) to estimate the system throughput. We analyze a *Recirculation* (REC) blocking policy and compare it with the benchmark *Wait-On-Spot* (WOS) policy for robots that are blocked. Finally, we compare the performance and operational costs of the vertical and horizontal system.

The rest of the paper is organized as follows. In Section 2, the literature review is presented. Section 3 describes the vertical system as well as the blocking policies for the vertical system. In Section 4, we model the systems as queueing networks and in Section 5 we present solution approaches and estimate the performance. In Section 6, the numerical analysis is performed. Specifically, we present the optimal configuration of the vertical system and compare the performance of the two blocking policies. Section 7 compares the horizontal and the vertical system for costs. Conclusions are given in Section 8.

2 Literature review

Several studies have developed analytical models to analyze the performance of the horizontal system. The articles can be categorized into three main groups.

Modeling AVS/R Systems: Malmberg (2002) was the first to analyze the horizontal (AVS/R) system. He developed a state equation model to estimate the vehicle utilization and cycle time as a function of system design parameters, e.g. the number shuttles, lifts, tiers and storage columns. Malmberg (2003) extended the state equation model by including the pending transactions in the state space description as well to estimate the storage and retrieval cycle time, system utilization and throughput capacity for various system design profiles. Note that the state equation approach is computationally inefficient for solving large scale problems. Therefore, Kuo et al. (2007) and Fukunari and Malmberg (2008) proposed a nested queueing approach to overcome this problem. In this approach, the lift is modeled as a queueing system which is nested within a separate vehicle queueing system. In the lift queueing system, vehicles are customers and the lifts are servers, while in the vehicle queueing system, transactions are customers and vehicles are servers. The two systems are analyzed iteratively until the performance measures converge. The problem with this approach is its inability to model a scenario in which the cycle starts from outside the storage rack, for example, when loads are received from outside the storage rack. To address this drawback, Fukunari and Malmberg (2009) proposed a queueing network model as an alternative to the nested queueing approach to estimate the performance of AVS/R system. This enabled them to also capture the performance of the system when interfacing with outside systems. Although these models are efficient in estimating the vehicle utilization with reasonable accuracy, they usually fail to estimate the transaction waiting time accurately. Zhang et al. (2009) addressed this problem by dynamically choosing among three different queueing approximations, based on the squared coefficient of variation (SCV) of transaction inter-arrival times. Using this procedure, they were able to increase the accuracy of the transaction waiting time estimation significantly. Heragu et al. (2011) used an open queueing network model and an existing tool called Manufacturing Performance Analyzer (MPA) to analyze the performance of the AVS/R system. Then they provided an extensive comparison between the performance of AVS/R systems and traditional AS/R systems. Marchet et al. (2012) also used the open queueing network approach to estimate the transaction cycle time of the AVS/R system with tier-captive vehicles. Ekren et al. (2013), Cai et al. (2014), and Roy et al. (2015b) used semi-open queueing networks to evaluate

the performance of the AVS/R system with tier-to-tier (pooled) vehicles. Lerher et al. (2015) derived a closed-form expression of the mean travel time for single and dual-command orders in a single-deep AVS/R system.

Design Choices for AVS/R Systems: Fukunari et al. (2004) studied the choice of vehicle dwell points in the AVS/R system, using a decision-tree analysis. Kuo et al. (2008) used the queueing network approach to investigate the effect of the class-based storage policy on the cycle time for an AVS/R system. Ekren and Heragu (2010) used simulation in combination with a regression analysis to analyze the effect of different rack configurations on the system performance. In the regression model, they investigated the effect of three inputs (number of tiers, aisles, and bays) on system performance measures. Ekren et al. (2010) developed a simulation-based experimental design to identify several factors that affect the performance of the AVS/RS. These factors include the L/U point location, dwell point policy, scheduling rule, and interleaving rule and their effect on the storage and retrieval transaction average cycle time, and average utilization of vehicles and lifts. Roy et al. (2012) modeled one tier of the AVS/R system as a multi-class semi-open queueing network with class switching to investigate the impact of design decisions on the performance of the system. They used a decomposition approach to evaluate the effect of different system depth to width ratios, vehicle assignment rules, and multiple storage zones on the expected system cycle time and vehicle utilization. Roy et al. (2015a) used a semi-open queueing network to analyze the optimal choice of dwell-point location and cross-aisle location. Their study showed that the end of the aisle is the optimal location for the cross-aisles, while the L/U point dwell policy improved the performance of the system.

Control Policies for AVS/R Systems: The more complex and stochastic nature of AVS/R system operation requires dynamic and real-time control policies. Especially, in some variants of the system with bi-directional single lanes, there is a possibility of a deadlock in the system. He and Luo (2009) used colored time Petri nets to dynamically model AVS/R systems to establish the necessary conditions to have a deadlock-free system. Roy et al. (2014) investigated the effect of vehicle blocking in the AVS/R system and proposed a semi-open queueing network to analyze the system performance and design trade-offs. Their results showed that the blocking delays

Table 1: Overview of the literature

| Research Category | Article | Research Issue | Methodology |
|-------------------|--|--|--------------------------------------|
| Modeling | Malmborg (2002, 2003) | Estimate vehicle utilization and cycle time | State equation model |
| | Kuo et al. (2007), Fukunari and Malmborg (2008) | Estimate vehicle utilization and cycle time | Nested queueing model |
| | Fukunari and Malmborg (2009) | Estimate vehicle utilization and cycle time interfacing material flow system | Closed-queueing network |
| | Zhang et al. (2009) | Estimating transaction waiting time | Variance-based nested queueing model |
| | Heragu et al. (2011) | Comparison with AS/RS | Open-queueing network and MPA |
| | Marchet et al. (2012) | Estimate transaction cycle time in tier-captive AVS/RS | Open-queueing network |
| | Ekren et al. (2013), Cai et al. (2014), Roy et al. (2015b) | Model AVS/R system with with pooled vehicles | Semi-open queueing network |
| | Lerher et al. (2015) | Estimate mean travel time | Closed-form solution |
| Design Choices | Fukunari et al. (2004) | Optimal dwell point location | Decision-tree analysis |
| | Kuo et al. (2008) | Effect of class-based storage on cycle time | Closed-queueing network |
| | Ekren and Heragu (2010) | Optimal Rack configuration | Simulation based regression |
| | Ekren et al. (2010) | Effect of design choices on performance | Simulation model |
| | Roy et al. (2012) | Effect of design choices on cycle time and vehicle utilization | Semi-open queueing network |
| | Roy et al. (2015a) | Optimal dwell-point and cross-aisle location | Semi-open queueing network |
| Control Policy | He and Luo (2009) | Deadlock-free control policy | Colored time Petri nets |
| | Roy et al. (2014) | Effect of vehicle blocking on performance | Semi-open queueing network |
| | Roy et al. (2016) | Effect of vehicle blocking on performance | Simulation model |
| | Zou et al. (2016) | Simultaneously vs sequentially requesting vehicles and lifts | Fork-join queueing network |
| | This Paper | Modeling the vertical system; effect of blocking on performance; comparison with the horizontal system | Closed-queueing network |

could contribute up to 20% of the transaction time. A detailed simulation model was developed by Roy et al. (2016) to evaluate the blocking delay in the AVS/R system. Zou et al. (2016) investigated a scenario in which the lift and vehicles in the tier-captive AVS/R system are requested simultaneously instead of sequentially. Using a fork-join queueing network, they showed that for a system with less than ten tiers, the parallel processing policy resulted in at least 5% performance improvement compared to the sequential processing policy. Table 1 presents the overview of all this literature.

The contribution of this study is threefold:

1. **Modeling:** We are first study to investigate *vertical* robotic based storage systems. In these systems, robots move horizontally as well as vertically to perform order transactions. This makes them fundamentally different from previous AVS/R systems. Therefore, new models are required to capture their performance.
2. **Methodological:** We present a jump-over approximation method to analyze the REC policy. This paper the first study to use this technique in a robotic warehousing context.
3. **Design insights:** We are the first study to investigate the optimal system size of the new generation robotic-based vertical system. We also compare the effect of two blocking policies

on the system throughput capacity. Furthermore, we are the first to compare cost as well as system throughput performance of vertical systems and of the commonly-known horizontal systems (AVS/RS).

3 System Description and Assumptions

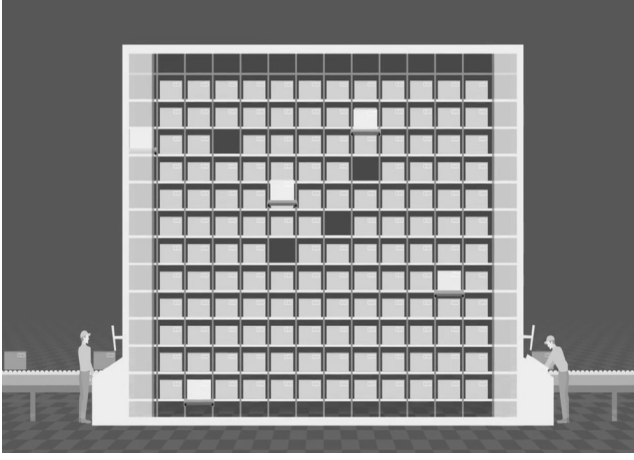
In Section 3.1, we describe the vertical system and our modeling assumptions. In Section 3.2, we present two blocking policies for the vertical system to investigate the effect of blocking on performance. Table 2 presents the main notations used in this study.

Table 2: Main notations

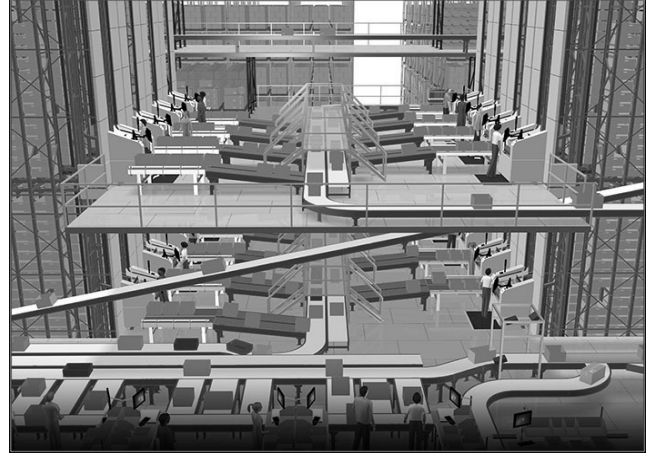
| Notation | Description |
|------------------|--|
| N_T | number of tiers |
| N_C | number of rack sections (columns) |
| n | total number of storage positions in one aisle of the system ($2N_TN_C$) |
| h | unit height clearance (tier height) |
| w | unit width clearance (rack section width) |
| K | number of robots (vertical) / shuttles (horizontal) |
| N_{LU} | number of L/U points in the system |
| v_r | velocity of the robots (vertical system) |
| v_s | velocity of the shuttles (horizontal system) |
| v_l | velocity of the lift (horizontal system) |
| τ_l, τ_u | load and unload time |
| τ_{LU} | processing time in the L/U point (we refer to this as picking time) |

3.1 Vertical Robotic Storage and Retrieval Systems

The vertical system consists of several aisles. Each aisle consists of two single-deep storage racks separated by an aisle in which autonomous robots can move. Each robot can access every storage location within the aisle by independently moving horizontally and vertically in sequence. The load and unload point (L/U point) is located at either end or at both ends of each aisle. Additional L/U points can be included on mezzanine floor to increase the pick capacity (see Figure 2). Technically, the robots have the ability to move between aisles. However, in current implementations of the vertical system, the robots are captive to an aisle. Therefore, in this study, we analyze a single



(a) Ground floor L/U point locations



(b) Mezzanine floor L/U point locations

Figure 2: L/U point locations in the vertical system (Source: Bastian Solution)

aisle of the vertical system. Figure 3 shows a side view of a single aisle of the vertical system with one L/U point. Each aisle is divided into several columns; we denote each column as a rack section. Robots follow a predefined direction path to access each storage location in the aisle. The outer loop is unidirectional while each rack section is bidirectional. Each time an order is placed

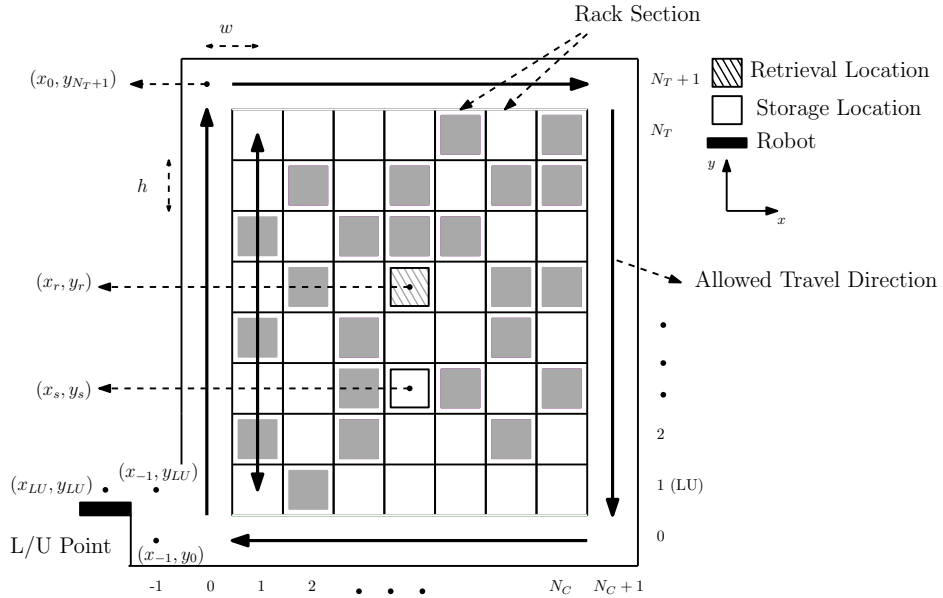


Figure 3: Side view of a single aisle

by a customer, a new retrieval request for a product is made. The control system dispatches the robot from its dwell point to the requested location, following the allowed direction. It picks up

the tote containing the ordered product and transports it to the L/U point. A similar process is followed to store a tote back into the storage location. The travel paths corresponding to the processing sequence of the storage and retrieval transactions in a dual-command cycle are depicted in Figures 4a and 4b respectively.

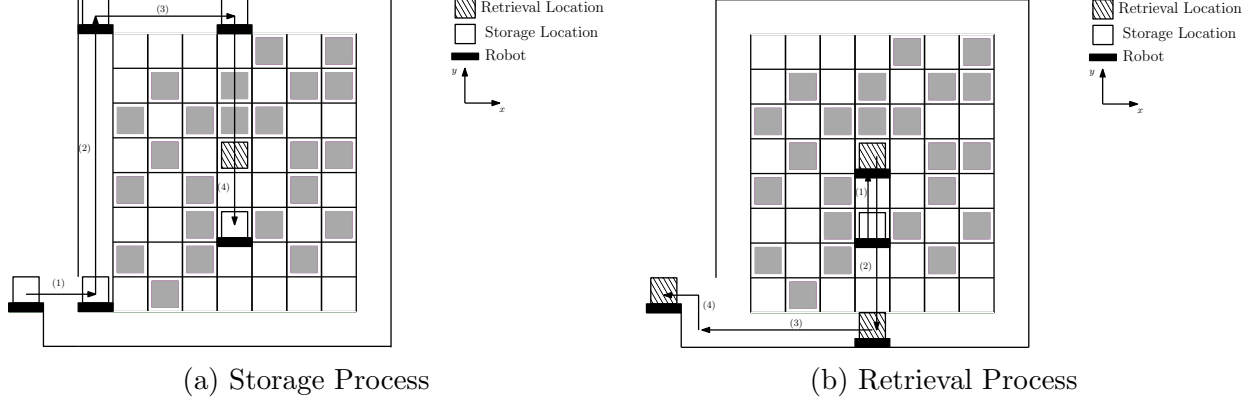


Figure 4: Dual-command cycle order

We make the following assumptions in analyzing the vertical system:

- a. *Dual-command cycle orders*: We assume the robot first stores the tote containing the previously fetched items and then retrieves the new tote containing items for the current request.
- b. *Same rack section*: We assume that the dual-command cycle orders always belong to a specific rack section, i.e. there is always a storage location available in the same rack section from which we want to retrieve the new item.
- c. *One L/U point*: We assume only one L/U point is available in each aisle unless stated otherwise (the models can be easily extended to accommodate multiple L/U points).
- d. *Uniform assignment*: We assume that storage and retrieval locations are assigned uniformly in the system, i.e. the probability of choosing any rack section as well as a location in any rack section is based on the uniform distribution.
- e. *One robot per rack section*: We assume only one robot can access each rack section at a certain time to avoid the risk of a deadlock in the system. Since each rack section has a bi-directional path.

- f. *Top entrance*: We assume robots enter each rack section from the top and exit from the bottom.
- g. *Constant velocity*: We ignore acceleration and deceleration of robots and use average velocity is in our analysis. We also ignored the driving direction changes since the changeover times are very short. The robots are also assumed to have the same velocity in both horizontal and vertical directions.

3.2 Effect of Blocking Delays on the Performance of the Vertical Robotic Storage and Retrieval Systems

To prevent a deadlock in the system, we assume that only one robot is allowed in each rack section at a certain time. If another robot wants to enter an occupied rack section, it needs to wait outside in the outer path, since there is no buffer location in a rack section for the waiting robots to queue. Consequently, increasing the number of robots can potentially lead congestion and blocking delays in the system. We call the policy in which the blocked robot waits on top of the rack section the WOS policy. The downside of this policy is that the waiting robot might block other robots that want to access another rack section, which negatively impacts the performance of the system. Therefore, we propose another waiting policy to mitigate this problem. In this policy, the REC policy, the robot first checks the status of the destination rack section and, if it is occupied, the robot circulates in the outer path around the rack sections. After completing one loop, it checks the status of the rack section again. If the rack section is no longer occupied, the robot claims it. Otherwise, it keeps recirculating until the rack section becomes available (Figure 5).

The REC policy can potentially result in a lower throughput, especially when the recirculation time is long. Furthermore, because the robots keep recirculating in the system, they consume more energy. However, by using this policy, we are sure that the robots do not block each other while waiting for the rack section to become empty, which can result in a higher throughput for a large number of robots.

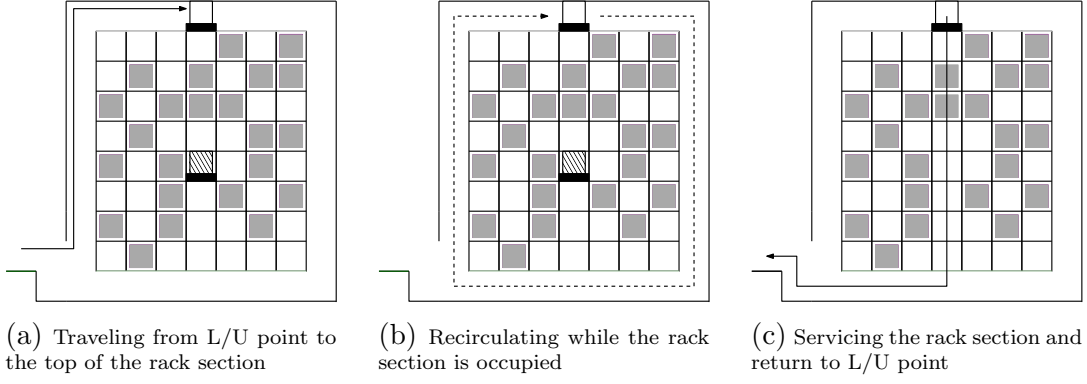


Figure 5: Recirculation policy

4 Vertical System Model Description

Increasing the number of robots in the system may cause delays. However, in Section 4.1, we first discuss a model where every rack section has unlimited buffer space. This ignores the blocking effect to establish the maximum system throughput capacity. Next, we investigate the effect of blocking delays on the performance of the system. In Section 4.2, we present the model for the WOS policy, and in Section 4.3, we discuss the model for the REC policy.

4.1 Unlimited Buffer Space inside each Rack Section

The process can be divided into three parts (see Figure 3): 1) Traveling to the top of the desired rack section: loading the item at the L/U point and transporting it from (x_{LU}, y_{LU}) to (x_s, y_{N_T+1}) . 2) The process within the rack section: descending from y_{N_T+1} to y_s and storing the tote, then going from y_s to y_r and retrieving the new tote, and finally descending from y_r to y_0 . 3) Transporting the item back to the L/U point: going from the bottom of the rack section (x_r, y_0) to the L/U point (x_{LU}, y_{LU}) and unloading the retrieved item tote. The whole process is illustrated in Equation 1.

$$\begin{aligned}
CT_{dc} = & \tau_{LU} + \frac{|x_{LU} - x_0|}{v_r} + \frac{|y_{LU} - y_{N_T+1}|}{v_r} + \frac{|x_0 - x_s|}{v_r} \\
& + \frac{|y_{N_T+1} - y_s|}{v_r} + \tau_u + \frac{|y_s - y_r|}{v_r} + \tau_l + \frac{|y_r - y_0|}{v_r} \\
& + \frac{|x_r - x_0|}{v_r} + \frac{|y_0 - y_{LU}|}{v_r} + \frac{|x_0 - x_{LU}|}{v_r}
\end{aligned} \tag{1}$$

In this section, we assume that each rack section has sufficient buffer locations for the waiting robots to queue. Figure 6 illustrates the corresponding closed-queueing network used to estimate the performance of the system.

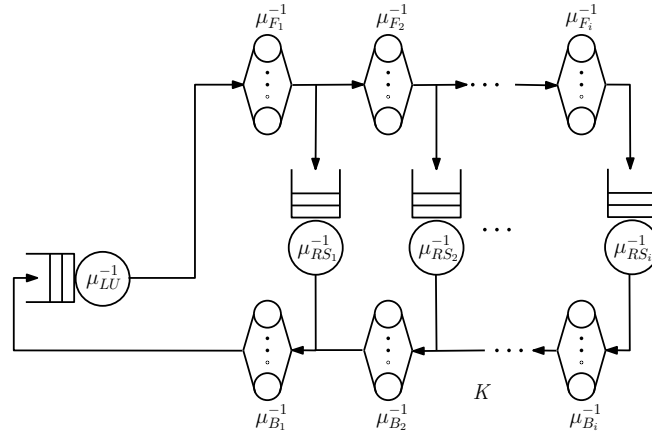


Figure 6: Closed-queueing network with infinite buffer

Infinite Server (IS) queues with deterministic service times are used to model travel to the top of the rack section (node F_i) and travel from the rack section to the pick station (node B_i). The L/U point is modeled as a single server queue with an exponential service time (node LU) and each rack section is modeled as a single server queue with a generally distributed service time (node RS_i). Although there is only one L/U point in the system, the network can also accommodate a second L/U point by extending it to a multi-class closed queueing network, in which robots are assigned to each L/U point based on their class.

Service Time Expression

Based on the system description and the assumptions, the expected service time of each node can be calculated. μ_{LU}^{-1} depends on the speed of the picker. Equations 2 and 3 show the derivations of $\mu_{F_i}^{-1}$ and $\mu_{B_i}^{-1}$, respectively in which $i = 1, \dots, N_C$ corresponds to the rack section number.

$$\mu_{F_i}^{-1} = \begin{cases} E \left[\frac{|x_{LU} - x_0|}{v_r} + \frac{|y_{LU} - y_{N_T+1}|}{v_r} + \frac{|x_0 - x_1|}{v_r} \right] = \frac{2 \times w}{v_r} + \frac{N_T \times h}{v_r} + \frac{w}{v_r} & , \text{ if } i = 1 \\ E \left[\frac{|x_i - x_{i-1}|}{v_r} \right] = \frac{w}{v_r} & , o.w \end{cases} \quad (2)$$

$$\mu_{B_i}^{-1} = \begin{cases} E \left[\frac{|x_1 - x_{-1}|}{v_r} + \frac{|y_0 - y_{LU}|}{v_r} + \frac{|x_{-1} - x_{LU}|}{v_r} \right] = \frac{2 \times w}{v_r} + \frac{h}{v_r} + \frac{w}{v_r} & , \text{ if } i = 1 \\ E \left[\frac{|x_i - x_{i-1}|}{v_r} \right] = \frac{w}{v_r} & , o.w \end{cases} \quad (3)$$

Equation 4 presents the travel time within RS_i . Note that y_s and y_r , the storage and retrieval locations within the rack section, are uniformly distributed.

$$T_{RS_i} = \frac{|y_{N_T+1} - y_s|}{v_r} + \tau_u + \frac{|y_s - y_r|}{v_r} + \tau_l + \frac{|y_r - y_0|}{v_r} \quad (4)$$

The expected value and squared coefficient of variation of the service time of RS_i is:

$$\mu_{RS_i}^{-1} = E[T_{RS_i}] \quad (5)$$

$$cv_{RS_i}^2 = \frac{Var[T_{RS_i}]}{[E[T_{RS_i}]]^2} \quad (6)$$

4.2 No Buffer Space inside a Rack Section and WOS Blocking Policy

In the WOS policy, the robots move to the top of the rack section and check whether the rack section is occupied. If it is not occupied, it moves to the rack section. If not, it waits on top. This process can be modeled using the blocking-after-service (BAS) protocol (Perros 1994), in which the robot either goes to the rack section node or waits at the previous node if the rack section is full. Figure 7 presents the corresponding queueing network.

In this network, all the nodes are modeled as single-server queues with exponentially distributed

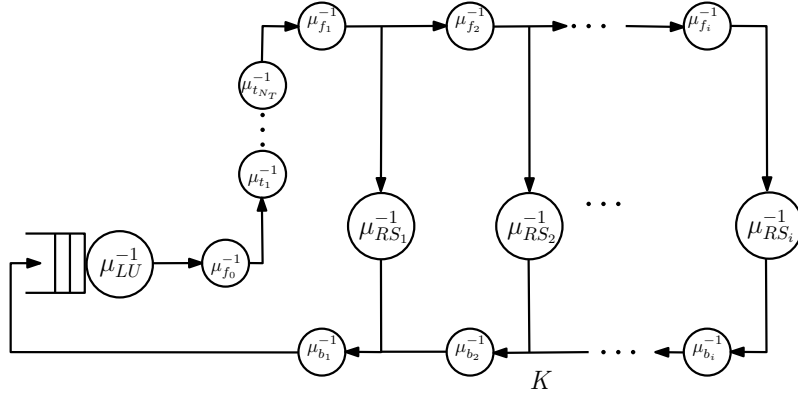


Figure 7: Closed-queueing network with finite buffer (WOS policy)

service times (this is needed for the solution approach to solve this network, which will be discussed in Section 5.2). The L/U point is assumed to have unlimited buffer space, while all the other nodes do not have any buffer space. Using an unlimited buffer at the L/U point is justified since in practice the robots can wait on a conveyor with ample space. The description of each node, as well as its expected service time, is presented in the next part.

Service Times Expressions

The expected service time in the L/U point and in the rack section do not change when including blocking in the system, therefore μ_{LU}^{-1} and $\mu_{RS_i}^{-1}$ have the same value as the unlimited buffer network presented in Section 4.1. $\mu_{f_0}^{-1}$ is the travel time to move horizontally from the L/U point to the location (x_0, y_1) in Figure 3, with expectation $\frac{2w}{v_r}$. $\mu_{t_i}^{-1}$ is the travel time to climb one tier, with expectation $\frac{h}{v_r}$. Similarly, $\mu_{f_i}^{-1}$ and $\mu_{b_i}^{-1}$ are the travel times to move horizontally the width of one RS, with expectation $\frac{w}{v_r}$; note that $i = 1, \dots, N_C$.

4.3 No Buffer Space inside a Rack Section and REC Blocking Policy

First, we build a network for one rack section, denoted by RS and then we extend it to the whole system. Upon service completion, the robot goes from the work station to the top of the rack section. This process is defined by two infinite server queues F and UP . If the rack section is not occupied, the robot enters the rack section. Otherwise, it recirculates in the system by going to the infinite server queue D which in combination with node UP creates the outer recirculation loop.

Note that the robot recirculates in the outer loop of the system (Figure 5b). After completing one loop, the robot checks the rack section again. Let p_b be the probability that the rack section is occupied. Then, with probability p_b , the robot goes to node D and with probability $1 - p_b$, the robot enters node RS . Upon service completion in RS , the robot continues its route to the L/U point via the infinite server queue G , which represents the travel time between the rack section and the L/U point. Figure 8 illustrates the resulting closed-queueing network.

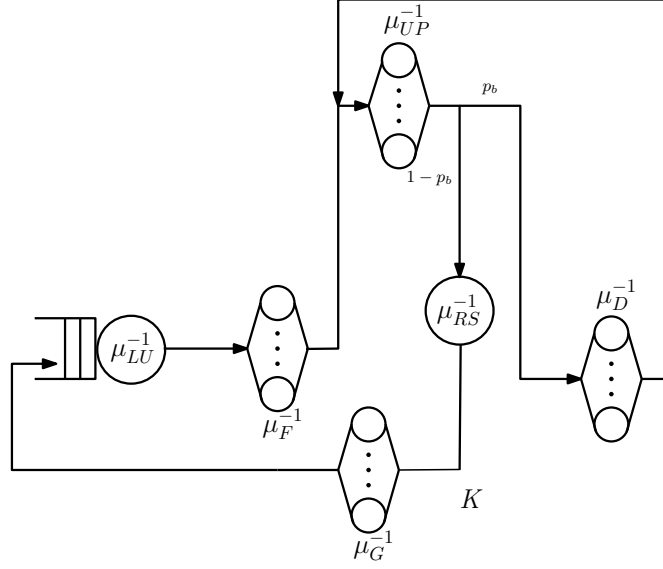


Figure 8: Closed-queueing network for one RS

Since the waiting robots circulate in the outer loop without stopping anywhere, we assume that no blocking occurs in the outer loop. Therefore, we can treat each rack section separately and extend the previous network to accommodate the rest of the rack sections. Figure 9 presents the resulting network.

Service Times Expressions

The terms, $\mu_{RS_i}^{-1}$ and μ_{LU}^{-1} have the same value as in Section 4.1. μ_F^{-1} is the travel time to move horizontally from the L/U point to the location (x_0, y_1) in figure 3. $\mu_{UP_i}^{-1}$ is the travel time to go from (x_0, y_1) to the top of RS_i , and $\mu_{B_i}^{-1}$ is the travel time to go from bottom of the RS_i to the L/U point. $\mu_{D_i}^{-1}$ is the travel time to go from the top of the RS_i to (x_0, y_1) , so that $\mu_{D_i}^{-1}$ plus $\mu_{UP_i}^{-1}$ creates the whole loop time. μ_F^{-1} , $\mu_{UP_i}^{-1}$, $\mu_{B_i}^{-1}$, and $\mu_{D_i}^{-1}$ are presented in Equations 7, 8, 9, and 10,

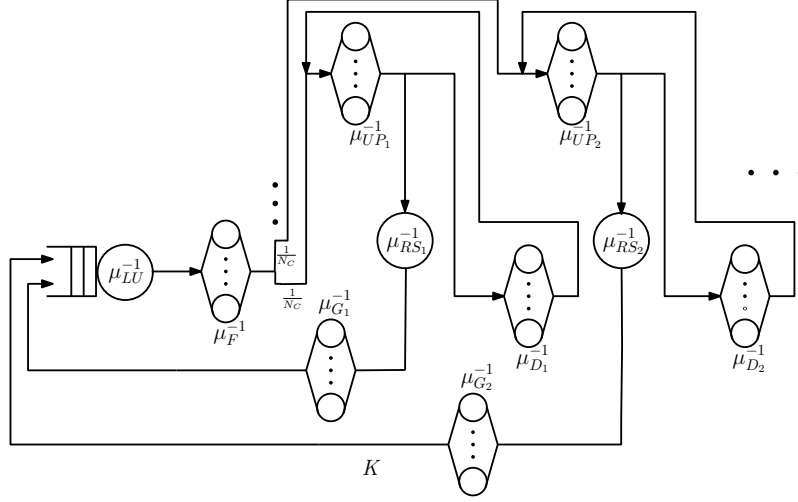


Figure 9: Closed-queueing network with finite buffer (REC policy)

respectively; note that $i = 1, \dots, N_C$.

$$\mu_F^{-1} = \frac{2 \times w}{v_r} \quad (7)$$

$$\mu_{UP_i}^{-1} = \frac{N_T \times h}{v_r} + \frac{(i) \times w}{v_r} \quad (8)$$

$$\mu_{G_i}^{-1} = \frac{i \times w}{v_r} + \frac{h}{v_r} + \frac{2 \times w}{v_r} \quad (9)$$

$$\mu_{D_i}^{-1} = \frac{((N_C + 1) - i) \times w}{v_r} + \frac{(N_T + 1) \times h}{v_r} + \frac{(N_C + 1) \times w}{v_r} + \frac{h}{v_r} \quad (10)$$

5 Solution Approach

In this section, we provide three solution approaches, to solve the networks that are presented in Section 4. Section 5.1 offers a solution to solve the network with unlimited buffer locations. Section 5.2 discusses the solution approach for solving the network for the WOS policy, and Section 5.3 presents a solution algorithm to solve the network corresponding to the REC policy.

5.1 Closed-Queueing Network with Unlimited Buffer

The RS_i nodes corresponding to the closed-queueing network described in Section 4.1 have generally distributed service times. Therefore the networks do not have a product form solution

(Baskett et al. 1975) and an approximation technique is required. We use Approximate Mean Value Analysis (AMVA) to approximate the performance of the networks. The AMVA method is an extension of the regular MVA method (Reiser and Lavenberg 1980), in which the squared coefficient of variation of the service time is used as a correction factor in calculating the residence times. The method is explained in Appendix A.

Validation

To validate the model for the vertical system, we compare the result with the output data provided by Bastian Solutions, a consulting company that has implemented such systems in practice. Table 3 presents the system parameters.

Table 3: System parameters (Bastian Solutions 2016)

| $n/2$ | w | h | K | v_r | τ_l, τ_u | τ_{LU} |
|-------|--------------|--------------|-----|--------------|------------------|--------------|
| 875 | 80 <i>cm</i> | 32 <i>cm</i> | 15 | 1 <i>m/s</i> | 1.5 <i>sec</i> | 5 <i>sec</i> |

The company only reported the maximum velocity of robots without accounting for acceleration/deceleration, to be 1.89 *m/s*. We adopt the approach by Lerher et al. (2010) to obtain the average velocity of the robot (see Appendix B). We assume that the robot has an acceleration/deceleration of 1 *m/s*² and calculate the average velocity of the robot inside the rack section. The resulting average velocity of the robot is 1 *m/s* which we use in the rest of this study. To estimate w and h , we have added 5% to the tote dimensions as reported by the company, to account for the rack size and other clearances in the rack structure. The system also has two L/U points, one at each end of the aisle. Using the given parameters, the system achieves a throughput of 800 dual-command cycles per hour according to the company. Using our analytical model we estimate the system throughput to be 834 dual-command cycles per hour (about 4.25% higher than what is achieved). This difference is mainly due to the absence of exact system parameters, as well as the real-time warehouse management and control protocols present in the vertical system.

5.2 Closed-Queueing Network without Buffer Location - WOS Policy

Due to the finite buffers, the queueing network developed in Section 4.2 does not have a product form solution (Perros 1994) and an approximation is needed to obtain the performance statistics. We use the approximation proposed by Akyildiz (1988) because it fits our network's characteristics. The approximation is a modified MVA for calculating the mean residence time of the jobs in the blocked node. First, the number of jobs in the destination node is calculated using MVA. If the calculated number of jobs is bigger than the number of buffer spaces in the destination node, the new job is blocked. Therefore, the new job remains in the source node until a spot in the destination node becomes available. This place will be available after a job has finished service at the destination node. Accordingly, the mean residence time of the new jobs in the source node increases by the mean remaining service time of the job in the destination node. The remaining service time of the job in the destination node is approximated to be the same as the expected service time of the destination node. The detailed explanation of the algorithm and its accuracy evaluation can be found in Akyildiz (1988).

5.3 Closed-Queueing Network without Buffer Location - REC Policy

The queueing network developed in Section 4.3 does not have a product form solution either. To accurately estimate the performance statistics, we approximate the network by another one with the jump-over blocking protocol (Van Dijk 1988). The jump-over network has a product form solution, and its performance statistics can be calculated using MVA (Van der Gaast et al. 2016). We first present how the jump-over network is built, and then we illustrate the MVA based algorithm to solve the network. Table 4 presents the notations used in explaining the jump-over network and the solution algorithm.

Jump-over Network

Assume a robot that intends to visit RS_i is labeled as either *entered* RS_i or *skipped* RS_i . Technically, this labeling is done based on the robot actually visiting the RS_i , i.e., it is labeled *entered*

Table 4: Notations for jump-over network

| <i>Notation</i> | <i>Description</i> |
|-----------------|---|
| k | number of robots in the network |
| K | total number of robots in the system |
| j, m, n | nodes in the network: $LU, F, UP_i, RS_i, D_i, B_i$ |
| i | rack section index: $1, 2, \dots, N_C$ |
| $T_j(k)$ | mean residence time of node j when there are k jobs in the network |
| $L_j(k)$ | mean number of robots at node j when there are k jobs in the network |
| $X(k)$ | overall throughput of the system when there are k jobs in the network |
| $P_{m,n}$ | routing probability of going from node m to node n |
| p_{b_i} | blocking probability of node RS_i |
| $U_j(k)$ | utilization of node j when there are k jobs in the network |
| $\pi_j(q k)$ | marginal probability of having q jobs in node j when there are k jobs in the system |
| V_j | visit ratio for node j |
| μ_j^{-1} | expected service time of node j |

RS_i if it actually entered RS_i and served, and it is labeled *skipped* RS_i if it skipped RS_i because the rack section was occupied. However, in the jump-over network, the robot goes in the rack section, but the labeling is done randomly and regardless of whether the robot was actually served at RS_i or not. The probability of a robot receiving either one of the labels is taken as the fraction of robots in the original network receiving the specific label, i.e., the fraction of robots that are labeled as *skipped* RS_i equals the blocking probability of RS_i in the original network (p_{b_i}). Initially, p_{b_i} is not known; however, it can be estimated iteratively using any initial guess. Hence, it is assumed that p_{b_i} is known beforehand in the jump-over network. Therefore, P_{RS_i, D_i} (routing probability to go from RS_i to D_i) equals p_{b_i} and P_{RS_i, G_i} (routing probability to go from RS_i to G_i) equals to $1 - p_{b_i}$. This means that the robot is labeled *skipped* RS_i and routed to the node D_i with probability p_{b_i} , and it is labeled *entered* RS_i and routed to the node G_i with probability $1 - p_{b_i}$. Finally, after the robot comes out of the RS_i , it follows the mentioned Markovian route regardless of whether it skipped or served inside RS_i . Figure 10 presents the corresponding network with the jump over protocol.

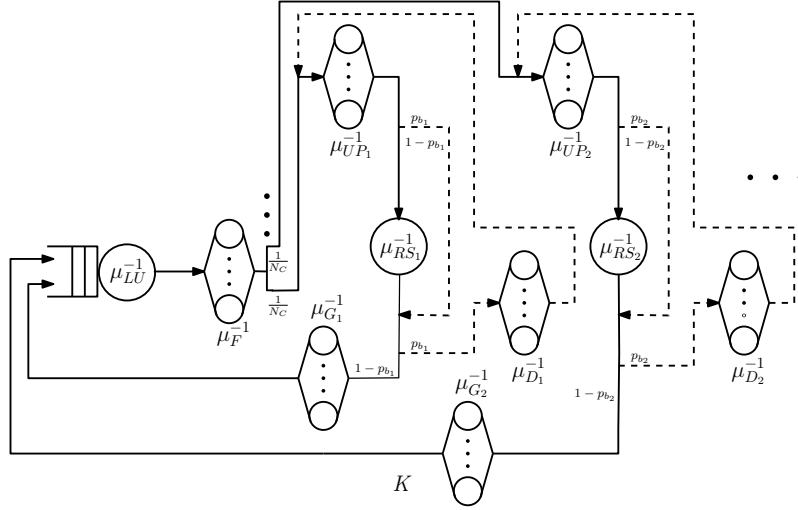


Figure 10: Jump-over approximation of the closed-queueing network in Figure 9

Visit Ratios

The visit ratios (V_j) can be calculated for each node according to the routing probabilities. We illustrate the approach to obtain visit ratios with one rack section in the system. Then, using the same idea, the visit ratio of the system with more than one rack section can be obtained. For that purpose, we follow a robot doing a dual-command cycle route, starting from the L/U point and coming back to the L/U point. Depending on how many times the robot has jumped the rack section, the number of times it visited each node in one route is different. Table 5 presents the number of times the robot visits each node on its route for different scenarios, along with the probability of a scenario.

Table 5: Number of times the robot visits each node

| LU | F | UP | RS | D | G | Probability |
|------|-----|------|------|-----|-----|------------------|
| 1 | 1 | 1 | 1 | 0 | 1 | $1 - p_b$ |
| 1 | 1 | 2 | 2 | 1 | 1 | $(1 - p_b)p_b$ |
| 1 | 1 | 3 | 3 | 2 | 1 | $(1 - p_b)p_b^2$ |
| 1 | 1 | 4 | 4 | 3 | 1 | $(1 - p_b)p_b^3$ |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |

The number of times the robot visits LU , F and G in a dual-command cycle is always 1. Therefore, the expected number of visits for LU , F and G is 1 ($V_{LU} = 1$, $V_F = 1$, $V_G = 1$). It is

clear from the Table 5, that the number of times the robot visits UP or RS follows a geometric distribution with parameter $1 - p_b$. Therefore, the expected number of visits for UP and RS is $V_{UP} = V_{RS} = \frac{1}{1-p_b}$. The number of times the robot visits D also follows the same geometric distribution, but starting from 0. Hence, the visit ratio for D is $V_D = \frac{p_b}{1-p_b}$. Similarly, the visit ratios for a network with more than one rack section can be obtained by multiplying them with the probability that the RS_i is selected.

Mean Value Analysis for Jump-over Network

The idea is to initialize the p_{b_i} with any arbitrary value, then to solve the jump-over network using MVA, update the blocking probabilities and repeat the MVA again until the difference between the current calculated blocking probability and the previous one is less than a given ϵ .

In each iteration of the MVA, given k robots in the system, the algorithm calculates $T_j(k)$, $X(k)$, $L_j(k)$, and $\pi_{RS_i}(1|k)$ (the marginal probabilities that RS_i is occupied). By iteratively increasing the number of robots in the system from 0 to K , we can calculate all the performance statistics. For the initialization phase ($k = 0$), we have $L_i(0) = 0$, $\pi_{RS_i}(0|0) = 1$, $\pi_{RS_i}(1|0) = 0$. Using the arrival theorem, the mean residence time of the LU , F , UP_i , D_i , and G_i are calculated as:

$$T_j(k) = \begin{cases} \frac{1}{\mu_j}(1 + L_j(k-1)) & \text{if } j = LU \\ \frac{1}{\mu_j} & \text{if } j = F, UP_i, D_i, G_i \end{cases} \quad (11)$$

The mean residence time of the RS_i is calculated by the following expression:

$$T_{RS_i}(k) = p_{b_i} \times 0 + (1 - p_{b_i}) \frac{1}{\mu_{RS_i}} \quad (12)$$

$$(13)$$

p_{b_i} is the probability that the RS_i is occupied, i.e. the probability of having one robot in RS_i when we have $k - 1$ robots in the system.

$$p_{b_i} = \pi_{RS_i}(1|k-1) \quad (14)$$

as a result:

$$T_{RS_i}(k) = \pi_{RS_i}(1|k-1) \times 0 + (1 - \pi_{RS_i}(1|k-1)) \frac{1}{\mu_{RS_i}} \quad (15)$$

Once the $T_j(k)$ s are obtained, the system throughput $X(k)$ is calculated by using Little's Law:

$$X(k) = \frac{k}{\sum_j V_j T_j(k)} \quad (16)$$

Using the throughput of the system and using Little's Law again, the new number of robots in each node for the next iteration is calculated:

$$L_j(k) = V_j X(k) T_j(k)$$

The marginal probabilities of having q robot in RS_i are calculated by balancing the number of transaction per time unit between state $q-1$ and q . The rate from q to $q-1$ is given by $\mu_{RS_i} \pi_{RS_i}(q|k)$, and by using the arrival theorem, the rate of going from $q-1$ to q is $V_{RS_i} X(k) \pi_{RS_i}(q-1|k-1)$. Since q is either 1 or 0 for RS_i we have:

$$\pi_{RS_i}(1|k) = \frac{V_{RS_i} X(k)}{\mu_{RS_i}} \pi_{RS_i}(0|k-1) \quad (17)$$

Furthermore, from the normalization constraint we have:

$$\pi_{RS_i}(0|k) = 1 - \pi_{RS_i}(1|k) \quad (18)$$

Therefore, the marginal probabilities can be calculated by exploiting Equations 17 and 18.

The performance statistics are estimated by sequentially applying the above equations. However, before the procedure starts, the blocking probability needs to be initialized. The initial value for the blocking probabilities can be any arbitrary number between zero and one. Based on this initial value for the blocking probabilities and by using the MVA method, the marginal probability of finding RS_i containing one robot is obtained. Then, we use this value as a new initial value for

the blocking probabilities and repeat the MVA again.

$$p_{b_i}^{new} = \pi_{RS_i}(1|K) \quad (19)$$

We continue the procedure until the difference between the newly calculated blocking probabilities and the current one is less than an ϵ .

$$|p_{b_i}^{new} - p_{b_i}^{current}| < \epsilon$$

The total procedure is presented in Appendix C.

Validation

We use simulation to see whether the jump-over approximation accurately estimates the performance statistics of the system with the REC policy. Instead of simulating the queueing network, we simulate a realistic implementation of an actual system to validate the model. The detailed simulation is built using the parameters listed in Table 3. The only difference is that the system has two rack sections and four tiers. The simulation is built in AutoModTM software version 12.3, simulation software commonly used by many material handling companies. The model uses the actual REC blocking policy and the exact physical dimensions of racks, totes, and robots, and exact travel times. Only the picking process is assumed to have an exponential distribution with unlimited buffer capacity. We run the simulation 40 times and record the results with 90% confidence intervals. Then, using the same parameters, we create the jump-over approximation of the system and solve it using the proposed algorithm. The blocking probabilities of each rack section and the system throughput of the system obtained by simulation and the analytical model are compared with two levels for the number of robots K in the system: five and 10. Table 6 illustrates the results.

The jump-over approximation results in less than 4% error, and is, therefore, an accurate representation of the original system with the REC blocking policy.

Table 6: Performance statistics

| K | Blocking Probability - RS_1 | | | Blocking Probability - RS_2 | | | System Throughput (DC/h) | | |
|-----|-------------------------------|--------|-------|-------------------------------|--------|-------|------------------------------|--------|-------|
| | Sim | Jump | Error | Sim | Jump | Error | Sim | Jump | Error |
| 5 | 0.4791 ± 0.0091 | 0.4626 | 3.44% | 0.4535 ± 0.0132 | 0.4657 | 2.69% | 683.12 ± 4.30 | 687.49 | 0.64% |
| 10 | 0.6393 ± 0.0081 | 0.6543 | 2.34% | 0.6787 ± 0.0053 | 0.6582 | 3.02% | 866.51 ± 6.41 | 895.98 | 3.40% |

DC/h : dual-command cycle per hour

6 Numerical Analysis

Choosing the right dimensions of the system directly influences the performance of the system. In Section 6.1, we investigate different layout configurations of the vertical system using the model developed in Section 4.1, to obtain the optimal layout. Later in Section 6.2, we compare the performance of the two blocking policies modeled in Sections 4.2 and 4.3 to evaluate under which conditions the blocking policies result in better system performance.

6.1 Optimal Configuration

Using the closed-queueing network developed in Section 4.1 (i.e. no blocking is assumed), we analyze different layout configurations, with varying height to width ratio (i.e., N_T/N_C), to obtain the optimal layout configuration. We investigate four levels of the number of storage locations ($n/2 = 300, 600, 900, 1200$) and two levels of the number of robots in the system ($K = 5$ and 10). The rest of the system parameters are the same as shown in Table 3. In each scenario, the system throughput is calculated while changing the number of tiers and rack sections in a way that the total number of storage location remains the same. The layout resulting in the maximum system throughput is identified as the optimal layout. Figure 11 illustrates system throughput for different scenarios. The x -axis is the value of $\frac{N_T}{N_C}$ in a logarithmic scale and the y -axis is the system throughput, measured in the number of dual-command cycles per hour.

Table 7 presents the layout configuration which achieves the maximum system throughput.

Since N_T and N_C take integer values, only certain ratios of N_T/N_C can be realized for a given number of storage locations. As a result, the presented optimal N_T/N_C ratios are not exactly the same for a different number of total storage locations. However, Table 7 and Figure 11 show that

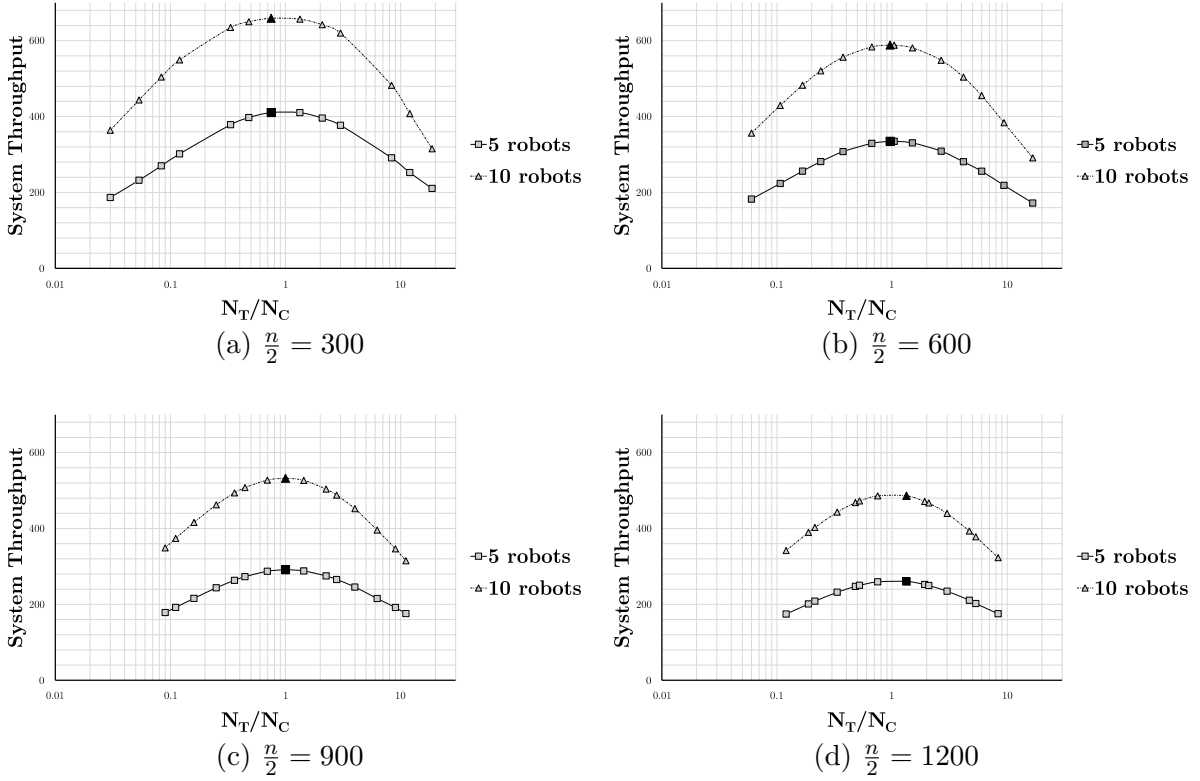


Figure 11: System throughput in different scenarios

Table 7: Optimal layout configurations

| N_s | R | N_T | N_C | $(N_T/N_C)^*$ | Throughput (DC/h) |
|-------|-----|-------|-------|---------------|-----------------------|
| 300 | 5 | 15 | 20 | 0.75 | 410.54 |
| 300 | 10 | 15 | 20 | 0.75 | 659.07 |
| 600 | 5 | 24 | 25 | 0.96 | 334.94 |
| 600 | 10 | 24 | 25 | 0.96 | 588.35 |
| 900 | 5 | 30 | 30 | 1.00 | 291.00 |
| 900 | 10 | 30 | 30 | 1.00 | 532.21 |
| 1200 | 5 | 40 | 30 | 1.33 | 260.45 |
| 1200 | 10 | 40 | 30 | 1.33 | 486.31 |

DC/h : dual-command cycles per hour

the optimal N_T/N_C ratios are close to 1 in each scenario.

Instead of using a queueing network, some studies adopt a probabilistic travel time approach to obtain the optimal layout configuration. In that method, the system configuration is chosen to minimize the dual-command throughput time. In Appendix D, this approach is presented for the vertical system. The results are presented in Table 8. The reported numbers are the feasible values for N_T and N_C while the numbers in parentheses are the actual numbers calculated by the

method.

Table 8: Optimal layout configurations using dual-command throughput time expression approach

| N_s | R | N_T | N_C | N_T/N_C | Throughput (DC/h) |
|-------|-----|------------|------------|-------------|-----------------------|
| 300 | 5 | 20 (17.92) | 15 (16.74) | 1.33 (1.07) | 410.36 |
| 300 | 10 | 20 (17.92) | 15 (16.74) | 1.33 (1.07) | 656.83 |
| 600 | 5 | 25 (25.35) | 24 (23.66) | 1.04 (1.07) | 334.80 |
| 600 | 10 | 25 (25.35) | 24 (23.66) | 1.04 (1.07) | 588.05 |
| 900 | 5 | 30 (31.05) | 30 (28.99) | 1.00 (1.07) | 291.00 |
| 900 | 10 | 30 (31.05) | 30 (28.99) | 1.00 (1.07) | 532.21 |
| 1200 | 5 | 40 (35.86) | 30 (33.46) | 1.33 (1.07) | 260.45 |
| 1200 | 10 | 40 (35.86) | 30 (33.46) | 1.33 (1.07) | 486.31 |

DC/h : dual-command cycle per hour

The optimal layout resulting from the travel time expression approach leads to almost the same system throughput as the queueing network. Hence, in this particular case, the waiting times in the L/U point and rack section queues in the network do not affect the optimal layout configuration.

6.2 Comparing the Blocking Policies

Two blocking policies were modeled in Sections 4.2 and 4.3. In this section, we aim to find out how the REC policy performs compared to the basic WOS policy. We analyze the performance of a system with four rack sections and eight tiers ($N_C = 4$, $N_T = 8$). In order to see the blocking effect on performance, the system needs to be the bottleneck and not the picker. Therefore, we assume to have four servers (pickers) at the single L/U point each with a picking time of 3.6 sec (which corresponds to current state-of-the-art picking stations that achieve 1000 dual-command cycles per picker per hour). The high picking speed at the L/U point prevents the picking process become the bottleneck. The rest of the system parameters are the same as in Table 3. In this scenario, we increase the number of robots in the system from one to 15 and observe the system throughput obtained by the two blocking policies. The results are illustrated in Figure 12. The x -axis represents the number of robots in the system and the y -axis is the system throughput, measured in dual-command cycles per hour.

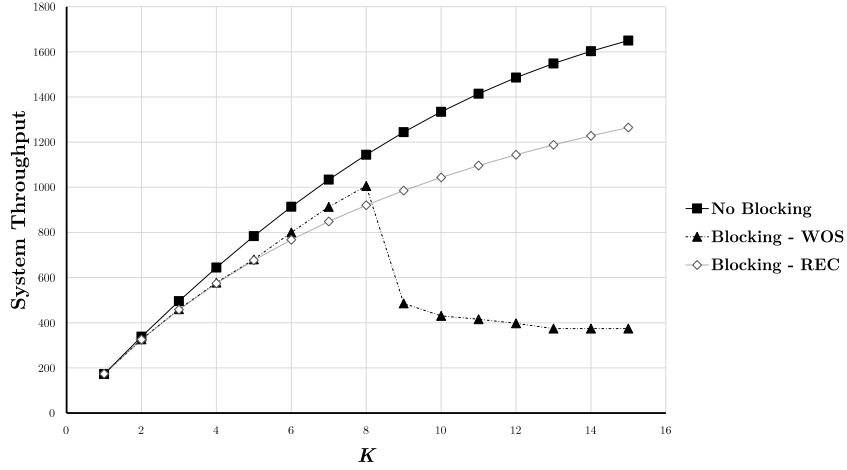


Figure 12: System throughput using different blocking policies

The graph (Figure 12) shows that the two policies perform approximately the same with a small number of robots in the system (the WOS policy has a slight advantage). However, if we increase the number of robots, the system throughput decreases sharply under the WOS policy. This decrease can be attributed to the waiting robots that block passing robots in the outer loop. The reason is that, there are some waiting positions in the system depending on the number of tiers and rack sections (see Figure 4.2), and once all the waiting positions are occupied, the blocking delays cause the system throughput to drop. In our example, the throughput drops if we increase the number of robots from eight to nine, since this can result in a situation in which all four rack sections are full with a robot waiting on top of each. Therefore, the ninth robot will definitely be delayed, resulting in a sudden decrease in the system throughput. This phenomenon is mitigated with the REC policy, in which robots are circulating in the system while waiting for their rack section to become available. Consequently, they never block other moving robots. If we compare the two policies, the REC has a lower system throughput, especially when the waiting time is less than the recirculation time and the number of robots is small. However, the REC policy can achieve a higher without worrying about the blocking delays when the number of robots in the system is increased. Therefore, the REC policy is a suitable blocking policy for the vertical system.

7 Cost-Performance Comparison of the Vertical and Horizontal System

Building on the previous sections, we now have the tools to compare the performance of the vertical and the horizontal systems. The horizontal system most comparable to the vertical system, is a system consisting of several aisles, where each aisle consists of two single-deep storage racks, with a dedicated lift and tier-to-tier shuttles. An L/U point is located at one end of each aisle (Figure 13). An extra L/U point can be added at the other end of the aisle, but then an extra lift needs to be installed as well. The lift can be either discrete or continuous. A discrete lift can only transport one shuttle at a time, while the continuous lift (i.e. a sort of vertical conveyor) can transport multiple shuttles simultaneously. However, the discrete lift is faster than a continuous lift. It should be noted that the shuttles can reach the first tier without needing a lift. However, a lift is needed to access the remaining tiers for the vertical movement. In this horizontal system, each aisle operates independently. Therefore, for a fair comparison, we analyze the performance of a single aisle of the vertical system with a single aisle of the horizontal system.

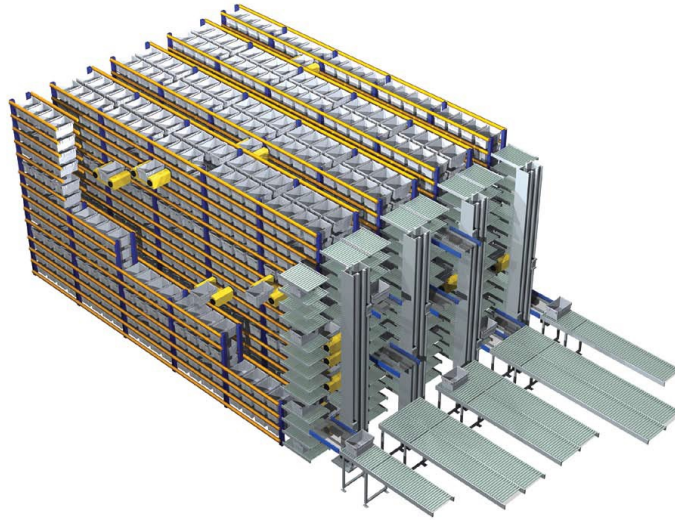


Figure 13: Horizontal system (Source: MultishuttleTM by Dematic)

To be in line with the vertical system analysis, we make similar assumptions for analyzing the horizontal system:

- a. *Dual-command cycle orders for shuttles:* We assume all requests are executed in a dual-command cycle.
- b. *Same tier:* We assume that the dual-command cycle orders are always for a specific tier. i.e. there is always a storage location available in the same tier from which we want to retrieve the new item.
- c. *Uniform assignment:* We assume that storage and retrieval locations are assigned uniformly in the system.
- d. *One shuttle per tier:* We assume only one shuttle can access each tier at a certain time in order to avoid the risk of having a deadlock in the system.
- e. *Constant velocity:* We ignore acceleration and deceleration of shuttles and discrete lift during movements in our analysis, and shuttles and lift have a constant velocity.
- f. *Dwell point:* We assume the Point of Service Completion (POSC) dwell point policy for the discrete lift. We assume the shuttles dwell at the L/U point.

In our analysis, we compare the maximum system throughput of the two systems for a given K , assuming there is enough buffer space available in each tier of the horizontal system and each rack section of the vertical system. Then we calculate the operating cost associated with the system throughput for each system to investigate which system achieves a higher throughput at a lower operating cost. This cost-performance analysis can guide decision makers to choose the right system based on performance needs and budget constraints. We estimate the performance of the horizontal system by developing a closed-queueing network model with unlimited buffer space, and estimate the performance of the vertical system by using the model developed in Section 4.1. The details of the model for the horizontal system are presented in Appendix E.

The operating cost is evaluated on the basis of the annualized cost as a sum of three items; equipment cost, floor space cost, and picker cost:

$$Cost = (C_s + C_r) \times K + C_l \times N_{LU} + C_A \times A + C_p \times N_{LU} \quad (20)$$

in which:

- C_s : annualized cost of a shuttle [€/year]
- C_l : annualized cost of a lift [€/year]
- C_r : annualized cost of a robot [€/year]
- C_p : annual cost of a picker [€/year]
- C_A : annual floor space cost [€/($m^2 \times \text{year}$)]
- A : floor space [m^2]
- K : number of robots or shuttles in the system
- N_{LU} : number of L/U points in the system

Note that C_s and C_l are zero for the vertical system, and C_r is zero for the horizontal system. The rack cost is assumed to be the same for both systems and therefore is ignored in our analysis. A is equal to $(w \times \text{depth}) \times N_C$ in both the vertical and horizontal system. Table 9 shows the parameters of both systems. The shuttles and the discrete lift velocity parameters are taken from MWPVL International (2013). The continuous lift velocity is obtained by analyzing a video of an implementation by Qimarox BV (2013). Different costs are listed in Table 10 (courtesy of Dynamis B.V, Marchet et al. (2012) and experts' opinions).

Table 9: System parameters

| System | $n/2$ | w | h | depth | v_r | v_s | $v_l(\text{disc})$ | $v_l(\text{cont})$ | τ_l, τ_u | $\tau_l, \tau_u(\text{lift})$ | τ_{LU} |
|-------------------|------------------|-------|-------|-------|-------|---------|--------------------|--------------------|------------------|-------------------------------|-------------|
| Vertical System | 300,600,900,1200 | 80 cm | 32 cm | 2.7 m | 1 m/s | N/A | N/A | N/A | 1.5 sec | N/A | 5 sec |
| Horizontal System | 300,600,900,1200 | 80 cm | 32 cm | 2.7 m | N/A | 3.2 m/s | 3.2 m/s | 1 m/s | 2 sec | 3 sec | 5 sec |

We assume that there is one L/U point available in the systems. We first determine the layout that maximizes the throughput, with a given storage capacity ($n/2 = 300, 600, 900, 1200$), given number of robots ($K = 1, \dots, 15$), and a system type. Then, the annualized cost of the configuration is calculate using the costs presented in Table 10, and assuming seven years of service and a 10% interest rate. Figure 14 illustrates the system throughput versus the annualized operating cost.

Table 10: Unit costs

| Cost item | Unit of measure | Value | Expected Life |
|-----------------|--------------------------------|--------|---------------|
| Shuttle | € | 15,000 | 7 years |
| Discrete Lift | € | 50,000 | 7 years |
| Continuous Lift | € | 60,000 | 7 years |
| Robot | € | 15,000 | 7 years |
| Floor Space | €/($m^2 \times \text{year}$) | 70 | N/A |
| Picker | €/year | 35,000 | N/A |

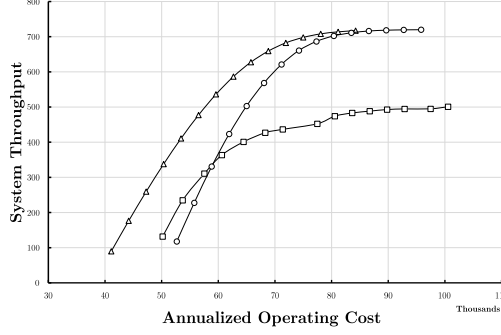
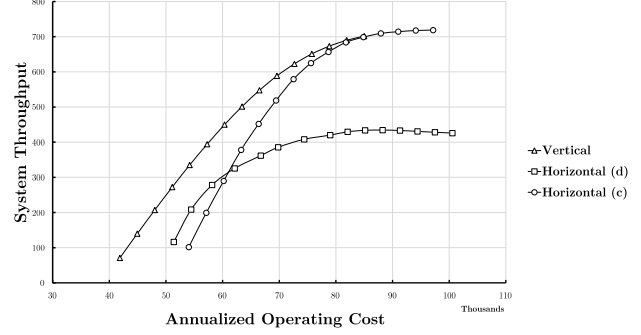
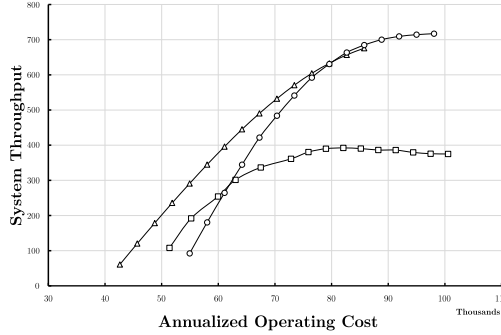
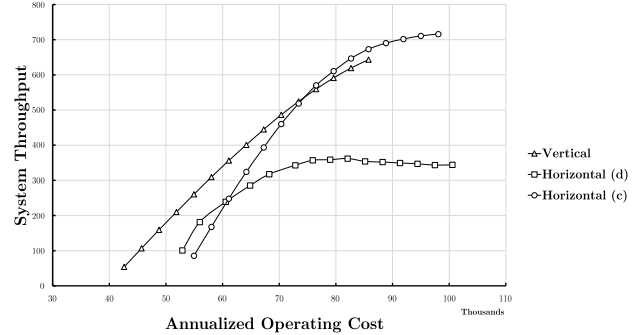
(a) $\frac{n}{2} = 300$ (b) $\frac{n}{2} = 600$ (c) $\frac{n}{2} = 900$ (d) $\frac{n}{2} = 1200$

Figure 14: System throughput vs. annual operating cost - one L/U point

The vertical system always outperforms the horizontal system with discrete lift (horizontal-d), both in terms of operating costs and system throughput. No matter what the number of storage locations in the system is, the vertical system always has lower operating costs compared to the horizontal-d system for the same throughput. Furthermore, the system throughput in the horizontal-d system saturates as the lift becomes the bottleneck in the system. Therefore, the upper bound for the throughput capacity depends on the capacity of the lift. For instance when $n/2 = 300$ the throughput capacity cannot be increased to more than 500 dual-command cycles per hour because the lift becomes the bottleneck at this point. Note that the capacity of the lift

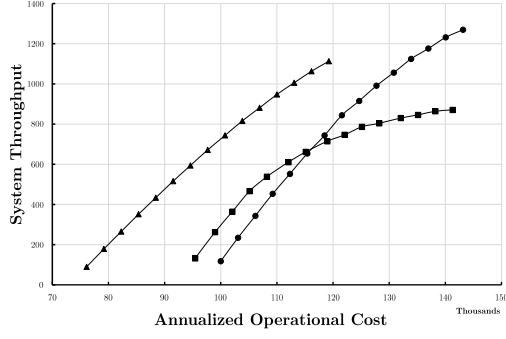
depends on the height of the rack, and since the rack is optimized for every individual instance, increasing the number of robots can lead to a different rack structure with different lift capacity. However, in the vertical system, the system throughput can be increased until the picker becomes the bottleneck, at 720 dual-command cycles per hour.

Compared to the horizontal system with continuous lift (horizontal-c), the horizontal-d system performs better when a lower system throughput is required. For a required system throughput of up to about 240 dual-command cycles per hour, the horizontal-d system has lower operating costs, while the horizontal-c system has a lower operating cost when a system throughput of more than about 390 dual-command cycles per hour is needed. Between a system throughput of 240 and 390, either system can perform with a lower operating cost depending on the number of storage locations in the system.

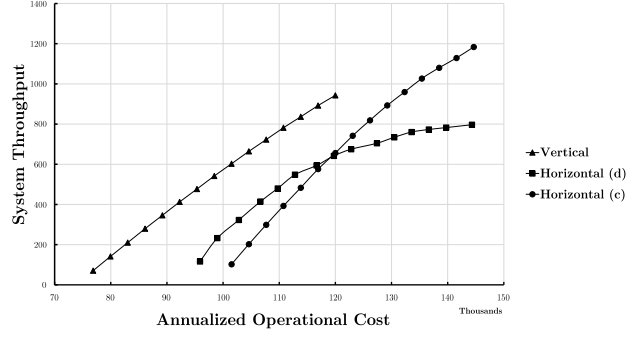
The performance of the vertical system also depends on the number of storage locations in the system. When $n/2 = 300$, the vertical system always produces a given system throughput at a lower operating cost compared to the horizontal-c system. However, as the number of storage locations increases, the operating cost of the vertical system increases more than the horizontal-c system. As a result when $n/2 = 1200$ the vertical system has a lower operating cost when the required system throughput is at most 540 dual-command cycles per hour. Beyond this threshold, the horizontal-c system has a lower operating cost.

It is possible to carry out the same analysis assuming two L/U points are available in each system (at either end of the aisle). The results are presented in Figure 15.

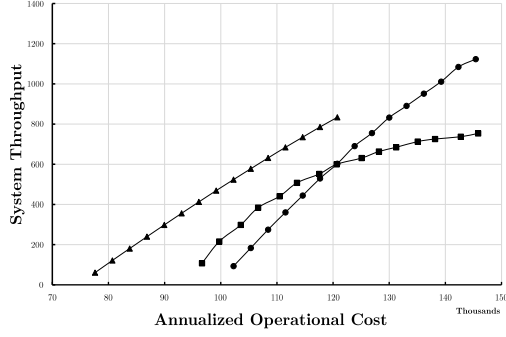
In this situation, the vertical system always has a lower operating cost compared to both horizontal systems. This is because an additional L/U point in the horizontal system requires the installation of an extra lift, which increases the operating cost significantly. For horizontal-c and horizontal-d systems, similar to the scenario with one L/U point, the operating cost depends on the number of storage locations in the system. When $n/2 = 300$, the horizontal-d system can generate a system throughput up to 380 dual-command cycles per hour, with lower operating cost compared to the horizontal-c system. The number decreases to 560 when $n/2 = 1200$. Similar to the one L/U scenario, the system throughput of the horizontal-d system saturates sooner than the



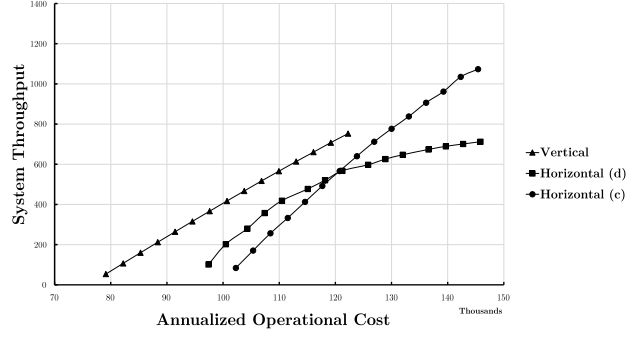
(a) $\frac{n}{2} = 300$



(b) $\frac{n}{2} = 600$



(c) $\frac{n}{2} = 900$



(d) $\frac{n}{2} = 1200$

Figure 15: System throughput vs. annual operating cost - two L/U points

other systems at around 1060 dual-command cycle per hour, since the lifts become the bottleneck.

Figure 16 summarizes the cost comparison of the three systems. In the figure, the annual operational costs of the vertical system (C_V), the horizontal-c system (C_{h-c}), and the horizontal-d system (C_{h-d}) are compared depending on the number of storage locations and the desired system throughput. Note that the throughput of the horizontal-d system cannot go beyond the black region in each scenario, since that is the point in which the lift becomes the bottleneck.

8 Conclusion

In this paper, we study a vertical storage and retrieval system. This is a new robotic-based system which eliminates the need for lifts for the vertical transportation. Robots in this system have the ability to move horizontally and vertically independently and access all the storage locations within the aisle. We develop analytical models to analyze the performance of the vertical system.

First, we assume there are infinite buffer locations available in each rack section and therefore

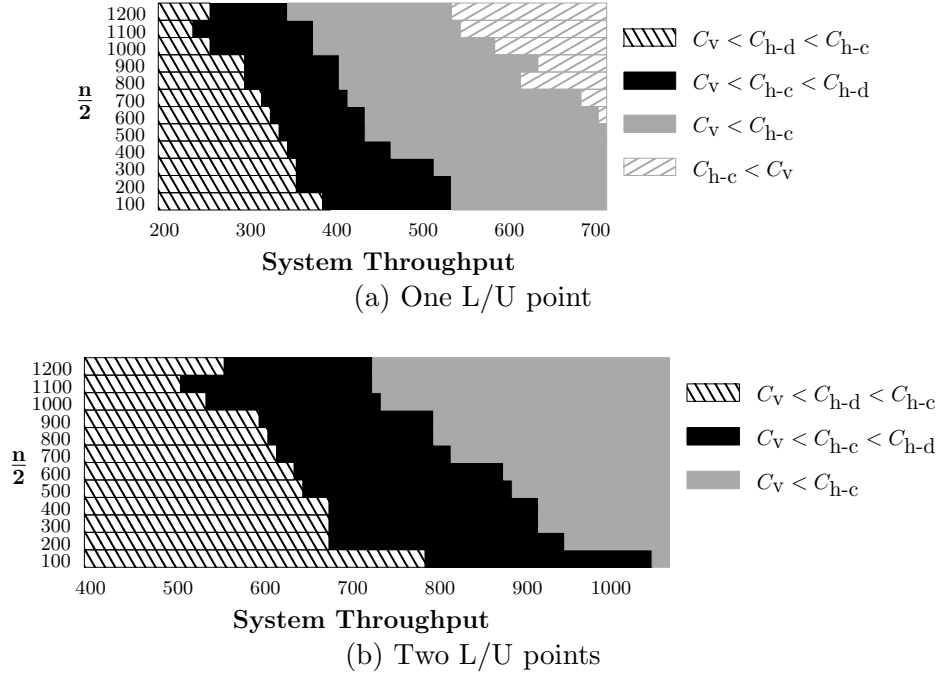


Figure 16: Summary of the cost comparison of the three systems

no blocking delays occur in the system. We develop a closed-queueing network and estimate the system throughput by using AMVA. We then develop a system dimension optimization model to obtain the optimal system layout for the maximum system throughput. The results show that when using the provided system parameters, the optimal ratio of height-to-width (i.e. number of tiers divided by the number of rack sections) is around 1. Next, we relax the assumption of having unlimited buffer locations in each rack section to estimate the effect of blocking delays on the system performance. We propose a REC waiting policy for the blocked robots, as opposed to the more obvious WOS policy in which the robots wait on top of the occupied rack section until it is unoccupied. We develop two closed-queueing networks corresponding to each policy to estimate the performance of the system. We use an approximate method proposed by Akyildiz (1988) to estimate the system throughput in the WOS network. In the case of REC network, We develop a new approximation technique by using the jump-over blocking protocol and an iterative algorithm based on MVA to estimate the performance of the system. This results in a very accurate estimation of the system throughput of the real system. Comparing the results of the two policies, the WOS policy has a slight advantage when the number of robots in the system is small. However, increasing the number of robots results in a sharp decrease in the system

throughput which can be mitigated by adopting the REC policy.

We then compare the performance of the vertical system with the horizontal system. The horizontal system is modeled by a closed-queueing network with an unlimited buffer and the system throughput is estimated using AMVA. We compare the cost-performance of the vertical system, the horizontal system with a discrete lift, and the horizontal system with a continuous lift. The results indicate that when there is one L/U point in the system, the vertical system outperforms the horizontal-d system in both operating costs and system throughput. However, compared to the horizontal-c system, its performance depends on the number of storage locations in the system.

When there are two L/U points in the system, the vertical system always has a lower operating cost compared to both the horizontal systems. Horizontal systems require an extra lift when there are two L/U points, which increases operating costs significantly.

For future studies, it would be interesting to examine the effect of different storage policies on the performance of the vertical system. Furthermore, it is interesting to investigate the effect of transaction sequencing policies on system throughput in the presence of multi-line orders.

References

- Akyildiz, I. F. (1988). Mean value analysis for blocking queueing networks. *IEEE Transactions on Software Engineering* 14(4), 418–428.
- Baskett, F., K. M. Chandy, R. R. Muntz, and F. G. Palacios (1975). Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM (JACM)* 22(2), 248–260.
- Bastian Solutions (2016). Personal communication.
- Bolch, G., S. Greiner, H. de Meer, and K. S. Trivedi (2006). *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons.
- Cai, X., S. S. Heragu, and Y. Liu (2014). Modeling and evaluating the avs/rs with tier-to-tier vehicles using a semi-open queueing network. *IIE Transactions* 46(9), 905–927.
- Ekren, B. Y. and S. S. Heragu (2010). Simulation-based regression analysis for the rack configuration

- of an autonomous vehicle storage and retrieval system. *International Journal of Production Research* 48(21), 6257–6274.
- Ekren, B. Y., S. S. Heragu, A. Krishnamurthy, and C. J. Malmberg (2010). Simulation based experimental design to identify factors affecting performance of avs/rs. *Computers & Industrial Engineering* 58(1), 175–185.
- Ekren, B. Y., S. S. Heragu, A. Krishnamurthy, and C. J. Malmberg (2013). An approximate solution for semi-open queueing network model of an autonomous vehicle storage and retrieval system. *IEEE Transactions on Automation Science and Engineering* 10(1), 205–215.
- Fraunhofer (2014). Rackracer - versatile and flexible rack feeder for container racks. Technical report, Fraunhofer Institute for Material Flow and Logistics.
- Fukunari, M., K. P. Bennett, and C. J. Malmorg (2004). Decision-tree learning in dwell point policies in autonomous vehicle storage and retrieval systems (avsrs). In *Machine Learning and Applications, 2004. Proceedings. 2004 International Conference on*, pp. 81–84. IEEE.
- Fukunari, M. and C. J. Malmberg (2008). An efficient cycle time model for autonomous vehicle storage and retrieval systems. *International Journal of Production Research* 46(12), 3167–3184.
- Fukunari, M. and C. J. Malmberg (2009). A network queuing approach for evaluation of performance measures in autonomous vehicle storage and retrieval systems. *European Journal of Operational Research* 193(1), 152 – 167.
- He, S. and J. Luo (2009). Deadlock control of autonomous vehicle storage and retrieval systems via coloured timed petri nets and digraph tools. *International Journal of Production Research* 47(12), 3253–3263.
- Heragu, S. S., X. Cai, A. Krishnamurthy, and C. J. Malmberg (2011). Analytical models for analysis of automated warehouse material handling systems. *International Journal of Production Research* 49(22), 6833–6861.
- Kuo, P.-H., A. Krishnamurthy, and C. J. Malmberg (2007). Design models for unit load storage and retrieval systems using autonomous vehicle technology and resource conserving storage and dwell point policies. *Applied Mathematical Modelling* 31(10), 2332–2346.
- Kuo, P.-H., A. Krishnamurthy, and C. J. Malmberg (2008). Performance modelling of autonomous vehicle

- storage and retrieval systems using class-based storage policies. *International Journal of Computer Applications in Technology* 31(3-4), 238–248.
- Lerher, T., B. Y. Ekren, G. Dukic, and B. Rosi (2015). Travel time model for shuttle-based storage and retrieval systems. *The International Journal of Advanced Manufacturing Technology* 78(9-12), 1705–1725.
- Lerher, T., I. Potrč, M. Šraml, and T. Tollazzi (2010). Travel time models for automated warehouses with aisle transferring storage and retrieval machine. *European Journal of Operational Research* 205(3), 571–583.
- Malmborg, C. J. (2002). Conceptualizing tools for autonomous vehicle storage and retrieval systems. *International Journal of Production Research* 40(8), 1807–1822.
- Malmborg, C. J. (2003). Interleaving dynamics in autonomous vehicle storage and retrieval systems. *International Journal of Production Research* 41(5), 1057–1069.
- Marchet, G., M. Melacini, S. Perotti, and E. Tappia (2012). Analytical model to estimate performances of autonomous vehicle storage and retrieval systems for product totes. *International Journal of Production Research* 50(24), 7134–7148.
- MWPVL International (2013). Dematic multishuttle 2, an in-depth and independent review from a supply chain consultant. Technical report.
- OPEX (2013). A unique approach to automated order fulfillment for small- to mid-sized ecommerce businesses. Technical report, OPEX Corporation.
- Perros, H. G. (1994). *Queueing networks with blocking*. Oxford University Press, Inc.
- Qimarox BV (2013). Prorunner mk5 vertical conveyor. <https://www.youtube.com/watch?v=e1uvFjKUZZ0>.
- Reiser, M. and S. S. Lavenberg (1980). Mean-value analysis of closed multichain queueing networks. *Journal of the ACM (JACM)* 27(2), 313–322.
- Roy, D., A. Krishnamurthy, S. S. Heragu, and C. J. Malmborg (2012). Performance analysis and design trade-offs in warehouses with autonomous vehicle technology. *IIE Transactions* 44(12), 1045–1060.
- Roy, D., A. Krishnamurthy, S. S. Heragu, and C. J. Malmborg (2014). Blocking effects in warehouse systems with autonomous vehicles. *IEEE Transactions on Automation Science and Engineering* 11(2), 439–451.

- Roy, D., A. Krishnamurthy, S. S. Heragu, and C. J. Malmberg (2015a). Queuing models to analyze dwell-point and cross-aisle location in autonomous vehicle-based warehouse systems. *European Journal of Operational Research* 242(1), 72–87.
- Roy, D., A. Krishnamurthy, S. S. Heragu, and C. J. Malmberg (2015b). Stochastic models for unit-load operations in warehouse systems with autonomous vehicles. *Annals of Operations Research* 231(1), 129–155.
- Roy, D., A. Krishnamurthy, S. S. Heragu, and C. J. Malmberg (2016). A simulation framework for studying blocking effects in warehouse systems with autonomous vehicles. *European Journal of Industrial Engineering* 10(1), 51–80.
- Tappia, E., D. Roy, R. D. Koster, and M. Melacini (2017). Modeling, analysis, and design insights for shuttle-based compact storage systems. *Transportation Science*, to appear.
- Van der Gaast, J. P., R. De Koster, I. Adan, and J. Resing (2016). Modeling and performance analysis of sequential zone picking systems. Working Paper.
- Van Dijk, N. M. (1988). On jackson’s product form with jump-overblocking. *Operations research letters* 7(5), 233–235.
- Zaerpour, N., Y. Yu, and R. B. de Koster (2015). Small is beautiful: A framework for evaluating and optimizing live-cube compact storage systems. *Transportation Science*.
- Zhang, L., A. Krishnamurthy, C. J. Malmberg, and S. S. Heragu (2009). Variance-based approximations of transaction waiting times in autonomous vehicle storage and retrieval systems. *European Journal of Industrial Engineering* 3(2), 146–169.
- Zou, B., X. Xu, and R. De Koster (2016). Modeling parallel movement of lifts and vehicles in tier-captive vehicle-based warehousing systems. *European Journal of Operational Research* 254(1), 51–67.

Appendix A Approximate Mean Value Analysis (AMVA)

Approximate Mean Value Analysis or AMVA is an approximation technique which extends the regular MVA (Reiser and Lavenberg 1980) to include nodes with a generally distributed service time. The MVA algorithm exploits two simple laws: a) *Little's law* which expresses the relation between mean residence time, throughput and mean number of jobs in a node of the overall network, and b) *Arrival theorem* which says that upon arrival at a node i , a job observes the system in the steady state with one job less (Bolch et al. 2006). Here we present the AMVA method for a single class closed-queueing network based on the MVA algorithm presented in Bolch et al. (2006). The notation used in the algorithm is presented in Table 11.

Table 11: Notations AMVA

| <i>Notation</i> | <i>Description</i> |
|-----------------|---|
| k | number of jobs in the network |
| K | total number of jobs in the system |
| i | nodes index: 1,2,...,N |
| j | servers index |
| $T_i(k)$ | mean residence time at node i when there are k jobs in the network |
| $L_i(k)$ | mean number of jobs at node i when there are k jobs in the network |
| $U_i(k)$ | utilization of node i when there are k jobs in the network |
| m_i | number of servers in node i |
| $X(k)$ | system throughput when there are k jobs in the network |
| $\pi_i(q k)$ | marginal probability of having q jobs in node j when there are k jobs in the system |
| V_i | visit ratio for node i |
| μ_i^{-1} | expected service time of node i |

The AMVA for a single class closed-queueing network is described as follows:

Step 1: Initialization. For all $i = 1, \dots, N$ and $j = 1, \dots, (m_i - 1)$, $E(L_i(0)) = 0$ and $\pi_i(0|0) = 0$ and $\pi_i(j|0) = 0$.

Step 2: Start iteration over the number of jobs $k = 1, \dots, K$

Step 3: Calculate the mean response time of a job at node i :

$$T_i(k) = \begin{cases} \frac{1}{\mu_i} & \times \left[1 + L_i(k-1) - U_i(k-1) \right] + U_i(k-1) \left(\frac{1+cv_i^2}{2} \right) & \text{for } m_i=1 \\ \frac{1}{\mu_i \cdot m_i} & \times \left[1 + L_i(k-1) - U_i(k-1) \right] + U_i(k-1) \left(\frac{1+cv_i^2}{2} \right) \\ & + \sum_{j=0}^{m_i-2} (m_i - j - 1) \cdot \pi_i(j|k-1) & \text{for } m_i > 1 \\ \frac{1}{\mu_i} & & \text{for } i = \text{IS} \end{cases} \quad (21)$$

The conditional probabilities are calculated by equations 22 and 23

$$\pi_i(0|k) = 1 - \sum_{j=1}^k \frac{X_i(k)}{\mu_i \cdot \alpha_i(j)} \cdot \pi_i(j-1|k-1) \quad (22)$$

$$\pi_i(j|k) = \frac{X_i(k)}{\mu_i \cdot \alpha_i(j)} \cdot \pi_i(j-1|k-1) \quad (23)$$

In which $\alpha_i(j)$ is:

$$\alpha_i(j) = \begin{cases} j & \text{if } j \leq m_i \\ m_i & \text{o.w} \end{cases} \quad (24)$$

Step 4: Compute the system throughput and throughput of each node using equations 25 and 26 respectively:

$$X(k) = \frac{k}{\sum_{i=1}^N V_i \cdot E[T_i(k)]} \quad (25)$$

$$X_i(k) = V_i \cdot X(k) \quad (26)$$

Step 5: Calculate mean number of jobs at node i using equation 27 and return to *Step 3* until $k = K$.

$$E[L_i(k)] = X_i(k) \cdot E[T_i(k)] \quad (27)$$

Appendix B Average Velocity Calculation

Since inside the rack section the robot stops twice to load and unload tote, its average velocity is much lower than its maximum velocity including the acceleration and deceleration. We use a procedure similar as Lerher et al. (2010) to calculate the average velocity of the robot. Depending on the storage and retrieval location, the robot's velocity either reaches its peak or not. Therefore, the velocity-time relationship of the robot is one of the graphs in Figure 17.

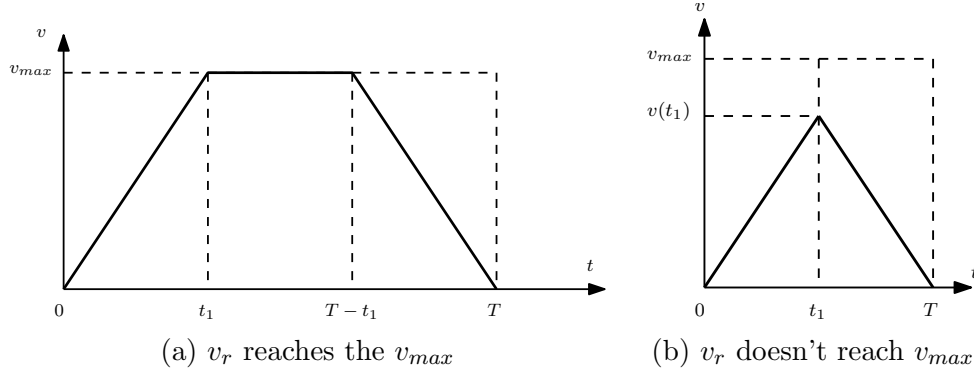


Figure 17: velocity-time relationship of the robot

We now present the average velocity for the robot in each scenario.

Velocity $v(t_1)$ is equal to v_{max} :

In this scenario, the velocity expression in dependence of time is presented in the following equation:

$$v(t) = \begin{cases} at & 0 < t < t_1 \\ v_{max} & t_1 < t < T - t_1 \\ -a(t - T) & T - t_1 < t < T \end{cases} \quad (28)$$

In which a is the acceleration. Now we can calculate the distance in depend of time by taking an integral from the velocity:

$$\begin{aligned} d(t) &= \int_0^t v(t)dt = \int_0^{t_1} atdt + \int_{t_1}^{T-t_1} v_{max}dt + \int_{T-t_1}^T -a(t - T)dt \\ &= \frac{at_1^2}{2} + v_{max}(T - 2t_1) + \frac{at_1^2}{2} \end{aligned} \quad (29)$$

It is clear from Figure 17a that $t_1 = v_{max}/a$. By substituting it in the equation 29 we have:

$$d(T) = v_{max} \cdot T - \frac{V_{max}^2}{a} \quad (30)$$

Therefore the average velocity is:

$$v_{avg} = \frac{d(T)}{T} = v_{max} - \frac{v_{max}^2}{T \cdot a} \quad (31)$$

Velocity $v(t_1)$ is less than v_{max} :

In this scenario, the velocity expression in dependence of time is presented in the following equation:

$$v(t) = \begin{cases} at & 0 < t < t_1 \\ -a(t - T) & t_1 < t < T \end{cases} \quad (32)$$

Now we can calculate the distance in depend of time by taking an integral from the velocity:

$$\begin{aligned} d(t) &= \int_0^t v(t)dt = \int_0^{t_1} atdt + \int_{T-t_1}^T -a(t - T)dt \\ &= \frac{at_1^2}{2} + \frac{a(T - t_1)^2}{2} \end{aligned} \quad (33)$$

It is clear from Figure 17a that $t_1 = T/2$. By substituting it in the equation 33 we have:

$$d(T) = \frac{a \cdot T^2}{4} \quad (34)$$

Therefore the average velocity is:

$$v_{avg} = \frac{d(T)}{T} = \frac{a \cdot T}{4} \quad (35)$$

Appendix C MVA for Jump-Over Network

Step 1: For all $i = 1, 2, \dots, N_C$, initialize the blocking probabilities: $p_{b_i}^{current} = \text{arbitrary number between } 0. \text{ and } 1.$

Step 2: For all $j = LU, F, UP_i, RS_i, D_i, B_i$ and $i = 1, 2, \dots, N_C$, initialize $L_j(0) = 0$, $\pi_{RS_i}(0|0) = 1$, $\pi_{RS_i}(1|0) = 0$.

Step 3: Start iteration over the number of jobs $k = 1, \dots, K$.

Step 4: Calculate mean throughput time for each node:

$$T_j(k) = \begin{cases} \frac{1}{\mu_j}(1 + L_j(k-1)) & \text{if } j = LU \\ \frac{1}{\mu_j} & \text{if } j = F, UP_i, D_i, B_i \\ \pi_j(1|k-1) \times 0 + (1 - \pi_j(1|k-1)) \frac{1}{\mu_j} & \text{if } j = RS_i \end{cases}$$

Step 5: Calculate the system throughput:

$$X(k) = \frac{k}{\sum_j V_j T_j(k)}$$

Step 6: Calculate mean queue length for each node:

$$L_j(k) = V_j X(k) T_j(k)$$

Step 7: Update marginal probabilities that the node RS_i is occupied:

$$(I) \pi_{RS_i}(1|k) = \frac{V_{RS_i} X(k)}{\mu_{RS_i}} \pi_{RS_i}(0|k-1)$$

$$(II) \pi_{RS_i}(0|k) = 1 - \pi_{RS_i}(1|k)$$

Step 8: Repeat *Steps 4 to 6*, until $k = K$

Step 9: Calculate new blocking probabilities for node RS_i :

$$p_{b_i}^{new} = \pi_{RS_i}(1|K)$$

Step 10: If $|p_{b_i}^{new} - p_{b_i}^{current}| > \epsilon$, update $p_{b_i}^{current} = p_{b_i}^{new}$ and return to *Step 2*, otherwise, end.

Appendix D Deriving Optimal Layout of the Vertical System by Using Travel Time Expressions

Taking the expected value from equation 1 and doing some simplifications results in equation 37:

$$\begin{aligned}
 E[CT_{dc}] = & \tau_{LU} + \frac{2.w}{v_r} + \frac{N_T.h}{v_r} + \frac{(N_C + 1).w}{2.v_r} \\
 & + \frac{(N_T + 1).h}{2.v_r} + \tau_u + \frac{(N_T + 1).h}{3.v_r} + \tau_l + \frac{(N_T + 1).h}{2.v_r} \\
 & + \frac{(N_C + 1).w}{2.v_r} + \frac{w}{v_r} + \frac{h}{v_r} + \frac{w}{v_r}
 \end{aligned} \tag{36}$$

$$\begin{aligned}
 E[CT_{dc}] = & \tau + \frac{w}{v_r} \left(2 + \frac{N_C + 1}{2} + \frac{N_C + 1}{2} + 1 + 1 \right) \\
 & + \frac{h}{v_r} \left(N_T + \frac{N_T + 1}{2} + \frac{N_T + 1}{3} + \frac{N_T + 1}{2} + 1 \right)
 \end{aligned} \tag{37}$$

in which

$$\tau = \tau_{LU} + \tau_u + \tau_l$$

The total storage number is $n = 2.N_T.N_C$. Therefore $N_C = \frac{n}{2.N_T}$. By substituting N_C in equation 37 we have,

$$E[CT_{dc}] = y(N_T) = \tau + \frac{w}{v_r} \left(5 + \frac{n}{2.N_T} \right) + \frac{h}{v_r} \left(\frac{7}{3}.N_T + \frac{7}{3} \right) \tag{38}$$

Taking the derivative of Equation 38 with respect to N_T and equaling it to 0 results in:

$$N_T = \sqrt{\frac{3.n.w}{14.h}} \tag{39}$$

$$N_C = \frac{n}{2.N_T} \tag{40}$$

Therefore:

$$\frac{N_T}{N_C} = \frac{3.w}{7.h} \quad (41)$$

Which in our case is, $\frac{N_T}{N_C} = \frac{3 \times 80}{7 \times 32} = 1.07$

Appendix E Horizontal System with Unlimited Buffer Locations inside each Tier

In Figure 18, the side view of a single aisle of the horizontal system is presented.

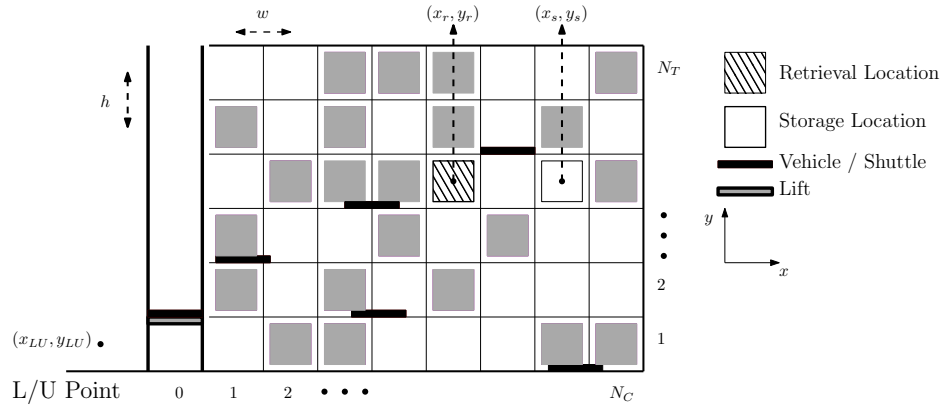


Figure 18: Side view of a single aisle of horizontal system

Figure 19a illustrate how to store an item in the horizontal system with a discrete lift. First, the shuttle waits for the elevator to pick it up (1). Next, the shuttle is transported vertically with the elevator in front of the desired tier (2). Then, the shuttles move horizontally inside the tier and stores the item (3). For the retrieval, which is illustrated in Figure 19b, first, the shuttle moves to the retrieval location inside the tier (1), picks the item and transports it to the front of the tier (2), and waits for the lift (3). The lift brings the shuttle down towards the L/U point for offloading the item (4). The movement of the shuttle is identical in the horizontal system with continuous lift, except that the shuttles do not wait for the lift and can immediately use it to go up or down.

Horizontal systems have been extensively studied in the literature (see Section 2). Here we

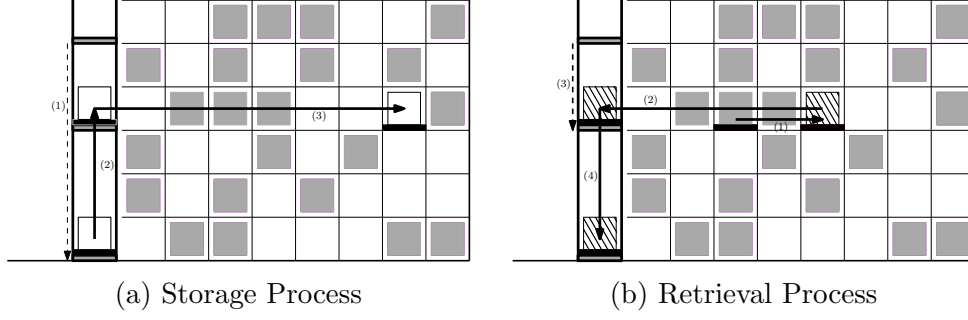


Figure 19: Storage and retrieval process in horizontal system

present the closed-queueing model we developed to estimate the performance of this system.

The horizontal system can use a discrete or a continuous lift for the vertical movement, each resulting in a slightly different model. Since the discrete lift can only serve one shuttle at a time, it is modeled as a single server queue with generally distributed service time. On the other hand, the continuous lift can serve multiple shuttles simultaneously; therefore, it is modeled as an infinite server queue.

Horizontal System with Discrete Lift

The L/U point is modeled as a single server queue with an exponentially distributed service time. Upon service completion at L/U point, the shuttle either directly goes to the first tier or use the lift to go to other tiers. Since tiers are uniformly selected, with probability $\frac{1}{N_T}$ the shuttle goes from the L/U point to the first tier, and with probability $\frac{N_T-1}{N_T}$, it uses the lift to go to other tiers. The load retrieval and storage process within each tier is modeled as a single server queue with generally distributed service time. Upon service completion in the tier, the shuttle is directed to the L/U point. If the shuttle is in the first tier, going back to the L/U point does not require using the lift. But if the shuttle is in any other tiers, it needs to use the lift to descend to the L/U point. The corresponding closed-queueing network is illustrated in Figure 20.

Service Times Expressions

μ_{LU}^{-1} depends on the speed of the picker. $\mu_{TR_i}^{-1}$ is based on the travel time expression for a shuttle to perform a dual-command in tier, $i = 1, 2, \dots, N_T$. The storage and retrieval locations in each

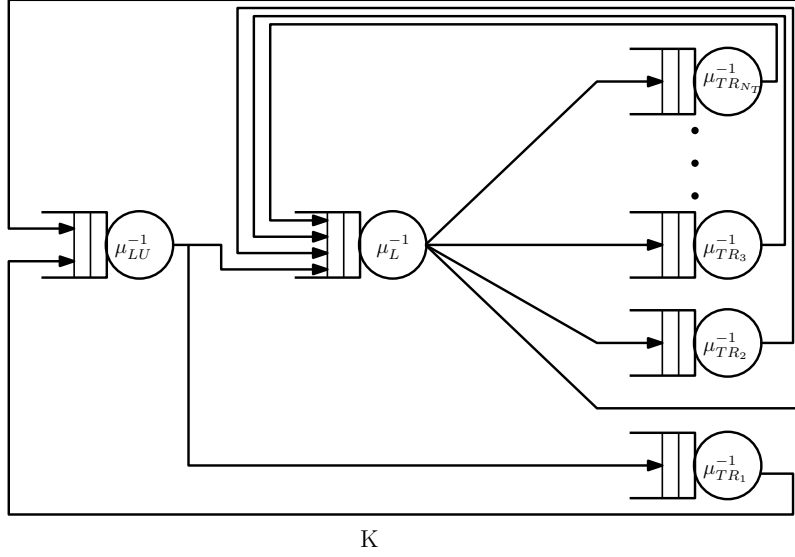


Figure 20: Closed-queueing network - the horizontal system with discrete lift

tier follow a uniform distribution from N_C available storage columns in each tier. Using Figure 18, the time expression to perform a dual-command cycle in tier i is presented in Equation 42.

$$T_{TR_i} = \frac{|x_0 - x_s|}{v_s} + \tau_u + \frac{|x_s - x_r|}{v_s} + \tau_l + \frac{|x_r - x_0|}{v_s} \quad (42)$$

Therefore, the expected service time and the squared coefficient of variation for the node TR_i is:

$$\mu_{TR_i}^{-1} = E[T_{TR_i}] \quad (43)$$

$$cv_{TR_i}^2 = \frac{Var[T_{TR_i}]}{[E[T_{TR_i}]]^2} \quad (44)$$

The time expression for the lift consists of two parts. We assume with probability $\frac{1}{2}$ the lift is doing a retrieval and with probability $\frac{1}{2}$ is doing a storage, i.e., half of the time the lift goes from its dwell position (y_d) to the L/U point to pick up a shuttle and to bring it to the desired tier (Equation 45), and half of the time the lift goes from its dwell position to another tier to pick up the shuttle and to bring it down to the L/U point (Equation 46).

$$T_{LS} = Y_{LS} = \frac{|y_d - y_{LU}|}{v_l} + \tau_l + \frac{|y_{LU} - y_s|}{v_l} + \tau_u \quad (45)$$

$$T_{LR} = Y_{LR} = \frac{|y_d - y_r|}{v_l} + \tau_l + \frac{|y_r - y_{LU}|}{v_l} + \tau_u \quad (46)$$

As we mentioned earlier, y_s and y_r are random variables with a uniform distribution between 2 and N_T . Also, the lift uses the POSC dwell policy, i.e., the lift dwells either at the L/U point in the first tier, after completing the retrieval process, or in front of a tier, after elevating up a shuttle to perform the storage process. Hence, y_d is a random variable, which with probability of $\frac{1}{2}$ is at tier 1 and with probability of $\frac{1}{2} \times \frac{1}{N_T-1}$ is at any other tier. So the service time of the lift has four scenarios as illustrated in Figure 21.

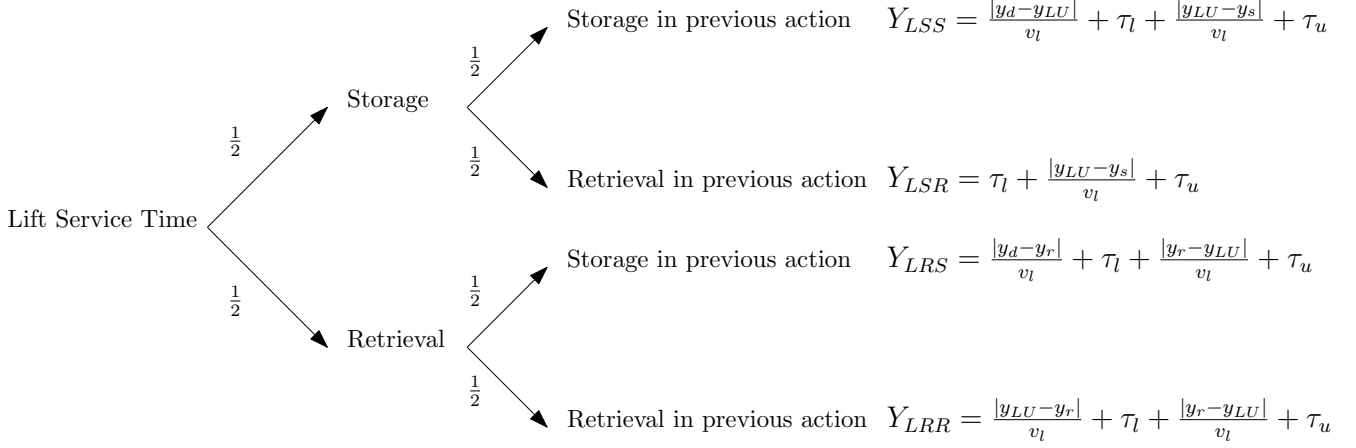


Figure 21: Lift service time

Y_{LSS} , Y_{LSR} , Y_{LRS} , and Y_{LRR} are random variables with probability density function $f_{LSS}(y)$, $f_{LSR}(y)$, $f_{LRS}(y)$, and $f_{LRR}(y)$ respectively. The total service time of the lift is a random variable Y_L with probability density function $g_L(y)$, which is equal to $f_{LSS}(y)$ with probability $\frac{1}{4}$, $f_{LSR}(y)$ with probability $\frac{1}{4}$, $f_{LRS}(y)$ with probability $\frac{1}{4}$, and $f_{LRR}(y)$ with probability $\frac{1}{4}$. The expected value of Y_L is calculated in Equation 47.

$$\begin{aligned}
E[Y_L] &= \int y g_L(y) dy \\
&= \frac{1}{4} \int y f_{LSS}(y) dy + \frac{1}{4} \int y f_{LSR}(y) dy + \frac{1}{4} \int y f_{LRS}(y) dy + \frac{1}{4} \int y f_{LRR}(y) dy \\
&= \frac{1}{4} E[Y_{LSS}] + \frac{1}{4} E[Y_{LSR}] + \frac{1}{4} E[Y_{LRS}] + \frac{1}{4} E[Y_{LRR}]
\end{aligned} \tag{47}$$

For calculating the variance of Y_L , first we calculate the expected value of Y_L^2 .

$$\begin{aligned}
E[Y_L^2] &= \int y^2 g_L(y) dy \\
&= \frac{1}{4} \int y^2 f_{LSS}(y) dy + \frac{1}{4} \int y^2 f_{LSR}(y) dy + \frac{1}{4} \int y^2 f_{LRS}(y) dy + \frac{1}{4} \int y^2 f_{LSR}(y) dy \\
&= \frac{1}{4} (Var[Y_{LSS}] + [E[Y_{LSS}]]^2) + \frac{1}{4} (Var[Y_{LSR}] + [E[Y_{LSR}]]^2) \\
&\quad + \frac{1}{4} (Var[Y_{LRS}] + [E[Y_{LRS}]]^2) + \frac{1}{4} (Var[Y_{LRR}] + [E[Y_{LRR}]]^2)
\end{aligned} \tag{48}$$

Now the variance is

$$Var[Y_L] = E[Y_L^2] - [E[Y_L]]^2 \tag{49}$$

Therefore the expected service time and the squared coefficient of variation for the node L is:

$$\mu_L^{-1} = E[Y_L] \tag{50}$$

$$cv_L^2 = \frac{Var[Y_L]}{[E[Y_L]]^2} \tag{51}$$

Horizontal System with Continuous Lift

As we described earlier, a continuous lift can provide service to more than one shuttles; therefore, the shuttle does not need to wait and can immediately utilize the lift for the vertical movement. Consequently, we can separate the storage and retrieval action of the lift, and model each as an infinite server queue. The rest of the network is similar to the discrete lift network. The resulting closed-queueing network is presented in Figure 22.

Service Times Expressions

The terms, μ_{LU}^{-1} and μ_{Ti}^{-1} take the same value as the one shown in the network for the system with discrete lift. The service time for the lift only depends on the original tier and the destination

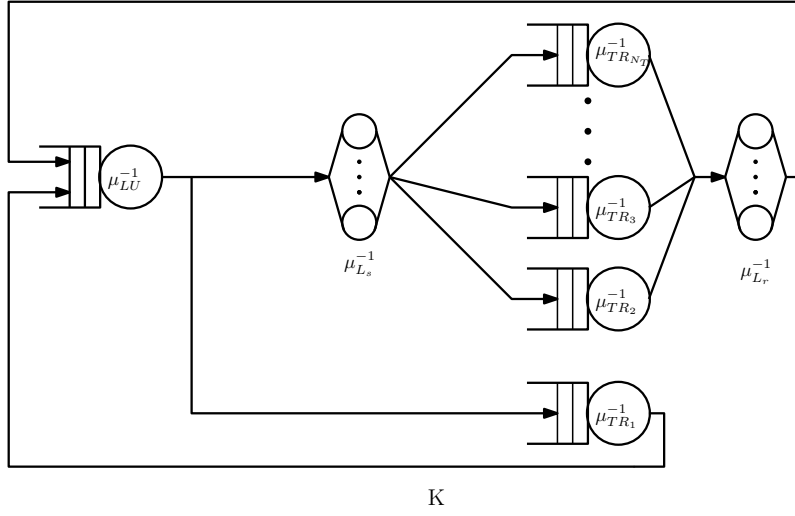


Figure 22: Closed-queueing network - the vertical system with continuous lift

tier, therefore,

$$\mu_{L_s}^{-1} = \mu_{L_r}^{-1} = \tau_l + \frac{1}{N_T - 1} \sum_{i=2}^{N_T} \left[\frac{1}{N_T - 1} \left(\frac{i-1}{v_l} \right) h \right] + \tau_u \quad (52)$$

Solution Approach

Since L and TR_i nodes in the closed-queueing network have generally distributed service times, we use AMVA to estimate the performance of the system.