Computers & Industrial Engineering 211 (2026) 111650

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A multi-queue single-server model for storage and retrieval requests in a tier-captive AVS/RS

Ilaria Battarra [a], Melike Baykal-Gürsoy [b], Riccardo Manzini [a,*]

[a] Department of Industrial Engineering, University of Bologna, Bologna, Italy
[b] Department of Industrial and Systems Engineering, Rutgers University, New Brunswick, NJ, USA

ARTICLE INFO

Keywords:
Automated warehousing
Autonomous vehicle storage and retrieval system
Analytical modelling
Performance analysis

ABSTRACT

Autonomous vehicle storage and retrieval systems (AVS/RSs) have become increasingly popular owing to their high storage density and throughput. Previous research had primarily focused on system performances with different layout configurations; the role of input and output buffers (bays) in the system tiers were largely overlooked. Vehicle-blocking events can occur when the bays are full, causing delays and degrading the performance. In this study, we propose a model that differentiates storage requests (which wait physically in the input bay) from retrieval requests (which remain in storage locations as in an 'electronic' queue). The aim of this study is to monitor the number of S-requests waiting in the input bay of the tier and the probability of vehicle-blocking events. This provides a valuable decision-support system for bay sizing in the warehouse pre design phase. Practical benefits include reduced number of blocking events, improved vehicle throughput, and guidance on recommended bay capacities. Although the proposed queueing model utilizes balance equations associated with a continuous-time Markov chain, we show that the performance results we seek are insensitive to the service-time distribution. A real-world case study provides insights into AVS/RS behaviour under various arrival rates and different dispatching policies.

1. Introduction

In today's global market, supply chains face significant challenges related to digitalisation, labour issues, and supply chain disruptions. Digitalisation transforms supply chains through advanced technologies such as the Internet of Things (IoT) and blockchain, which enhance visibility and streamline operations. Labour issues, including workforce shortages and the need for skilled labour, drive the shift towards automation. Additionally, supply chain disruptions, such as those caused by geopolitical tensions or pandemics, underscore the necessity for resilient and adaptive supply chain strategies. These factors are reshaping how companies manage and optimise their logistics networks.

In this complex network, actors must work collaboratively to design, produce, distribute, and deliver products or services to end consumers where and when requested. Time and flexibility have become the cornerstones of an efficient supply chain, and logistics play a pivotal role in coordinating the flow of goods. Specifically, warehouses act as strategic buffers between suppliers and customers. Traditional warehouses typically involve manual labour and conventional material-handling equipment (e.g., forklifts and basic shelves). Traditional warehouses have been essential in storage and logistics, but come with limitations. As supply chains grow more complex, the demand for automation and robotics in warehouses increases. Automation eliminates redundant operations, reduces the execution time of necessary warehouse activities, and increases safety (Kamali, 2019).

The autonomous vehicle storage and retrieval system (AVS/RS) is a cutting-edge warehouse-and-distribution-center management solution that revolutionizes how companies handle and store goods. This solution offers high storage density, flexibility, and short cycle time (Battarra et al., 2022a). The unit loads (ULs) handled are typically palletized ULs or small and light plastic containers (totes). According to the UL handled, the storage solution is called a shuttle-based storage and retrieval system (SBS/RS) for totes or an autonomous vehicle storage and retrieval system (AVS/RS) for palletized ULs. AVS/RS and SBS/RS are the hallmarks of automated storage systems because they allow parallel activities by assigning vertical and horizontal movements to vehicles such as lifts and shuttles. The resulting high flexibility overcomes the limits of other automated storage systems such as the automated storage and retrieval system (AS/RS), which uses stacker cranes to perform storage (S) and retrieval (R) requests.

* Corresponding author.
E-mail addresses: ilaria.battarra2@unibo.it (I. Battarra), gursoy@soe.rutgers.edu (M. Baykal-Gürsoy), riccardo.manzini@unibo.it (R. Manzini).

https://doi.org/10.1016/j.cie.2025.111650
Received 6 December 2024; Received in revised form 13 May 2025; Accepted 28 October 2025
Available online 1 November 2025
0360-8352/© 2025 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

In this study, we investigate a tier-captive single-aisle AVS/RS system. Fig. 1 illustrates the main elements.

The AVS/RS consists of n independent tiers connected by lifts (Battarra et al., 2024a). The lift that manages S-requests is called lift-in, whereas the one that manages R-requests is called lift-out. The ULs coming from the end-of-line area of a production/packaging system, i.e., the S-requests, enter the AVS/RS from the input buffer located at the ground tier (buffer-in, Fig. 1). The lift-in collects and delivers these ULs to the destination tier (y-direction). It can handle one or two ULs at a time, depending on its loading capacity. At the destination tier, the ULs arrive at the input buffer of the tier (bay-in, Fig. 1) via a conveyor system. The lift-out works in a similar manner, collecting the ULs from the output buffer on each tier (bay-out, Fig. 1) and delivering them to the output buffer at the ground tier (buffer-out, Fig. 1). Depending on the layout configuration, the bays at the system tiers can host a few ULs. In this case, a catenary system transports the ULs across the bay to the shuttle interface. The AVS/RS shown in Fig. 1 has double-capacity bays.

Each tier has a main double-sided cross aisle and multiple deep lanes. A multideep lane is a channel that can host two or more ULs, reaching depths of dozens of meters (Fig. 1C). Depending on the lane depth, each lane contains one or more ULs of the same item, called a stock-keeping unit (SKU). This storage policy, widespread in automated storage systems handling palletized ULs, is called the mono-SKU-lane storage policy (Lupi et al., 2024). A last-in-first-out (LIFO) policy is implemented for collecting the ULs, during retrieval.

Two independent vehicles, the shuttle and satellite, are assigned to each tier to serve storage and retrieval (S/R) requests. This is a tier-captive configuration. The shuttle moves back and forth along the main cross-aisle (x-direction). It transports the satellite and incoming UL to the destination lane or the outgoing UL to the bay-out (Fig. 1B). The satellite moves within the lane (z-direction) to collect or drop off the UL (Fig. 1C) and then rejoins the shuttle to continue the retrieval or storage cycle (this is the coupling time). They operate sequentially by simultaneously handling one UL. The S-requests wait for the shuttle at the bay-in of the tier, whereas the R-requests remain at their storage locations. The shuttle typically processes S/R requests according to a first-in first-out (FIFO) policy. However, alternative S/R dispatching policies can be adopted. In this study, these are called server policies because they involve allocating system resources (i.e., the shuttle) to S/R requests.

One of the most critical aspects of AVS/RSs is the expected system time, which is the time a request spends within the system, from arrival to service completion. This involves the time the UL travels within the storage system and the waiting time. Travelling and waiting times mainly depend on the vehicle's kinematics, system layout, and bay capacity. These times can be reduced by adjusting the vehicle's kinematics or adding new vehicles; however, modifying the layout and bay capacity can be challenging. In particular, the bay-in and bay-out capacities are shallow (e.g., they can host one or two ULs simultaneously) and are usually tied to the lifts' configurations, physical constraints of the building, or safety regulations. Two types of vehicle-blocking events can occur when a bay is full.

- When the bay-in is full, the lift-in can not bring the incoming ULs to the selected tier. The lift-in waits until at least one location in the bay becomes free. A vehicle-blocking event occurs whenever the lift waits.
- When the bay-out is full, the shuttle cannot collect other ULs. The shuttle waits until at least one location in the bay becomes free. A vehicle-blocking event occurs when no ULs can store or access the storage locations.

These vehicle-blocking events negatively affect the system performance. Measuring and monitoring the number of ULs waiting in the bays helps estimate the expected S/R time. It also helps track the probability of vehicle-blocking events and waiting times. These metrics provide crucial insights into the required bay capacity.

Designing efficient storage and retrieval systems is critical for optimizing warehouse operations, particularly in AVS/RS. One key aspect of this design is determining the appropriate size of the bays during the preliminary design stage. This decision directly impacts the space-utilization efficiency and system performance. Larger bays provide a higher capacity buffer, offering greater flexibility to the lift and shuttle systems by reducing congestion (Lupi et al., 2024, Battarra et al., 2022a, Battarra et al., 2024b). However, this comes at the expense of space that could otherwise be allocated for stock, potentially leading to underutilization of available storage.

Most studies on shuttle-based storage systems have presented analytical or simulation models to estimate the system performance in terms of the system throughput and cycle time. They typically couple S/R requests in a single queue (Eder, 2022; Ekren and Akpunar, 2021; Ekren et al., 2013; Lupi et al., 2023) or analyze the requests individually (Eder, 2019; Marchet et al., 2012). However, coupling S/R requests into

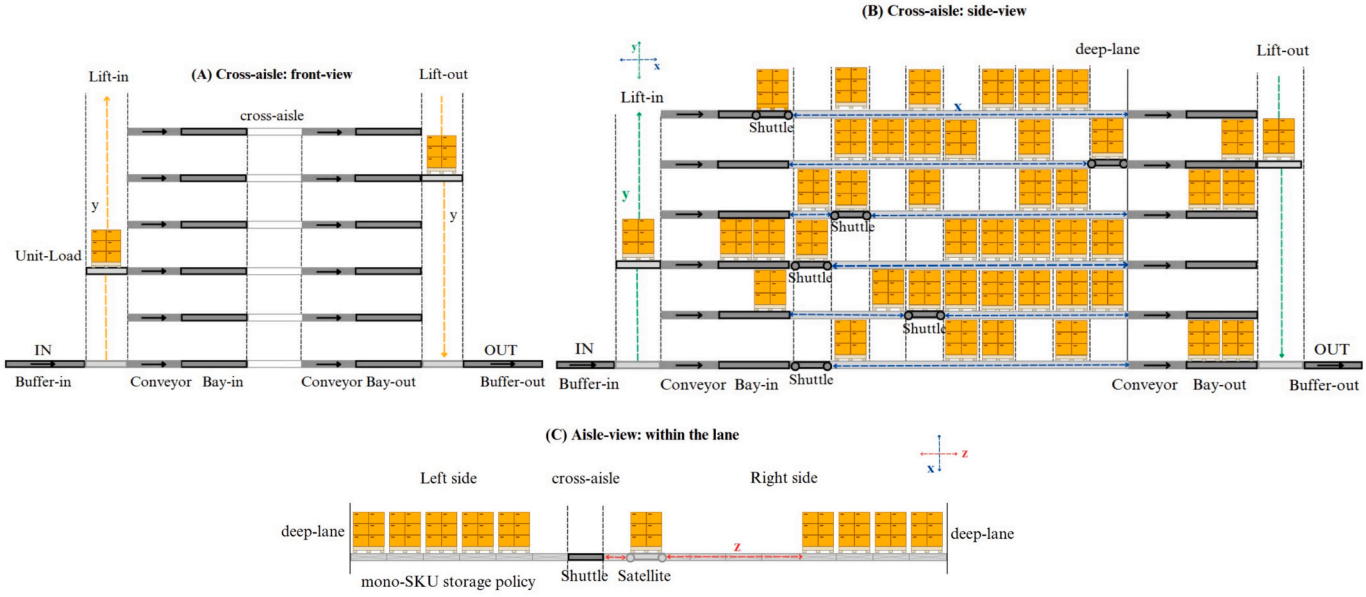


Fig. 1. Tier-captive AVS/RS with double capacity input and output bays: multi-tier and single-tier views.

a single queue limits the ability to monitor the filling degree of the bay-in area accurately. This is because storage and retrieval requests are associated with different physical waiting locations: storage requests physically queue at the bay-in, whereas retrieval requests remain at their original storage positions until served (i.e., in a virtual or electronic queue). Without separating storage and retrieval requests into different queues, only observation of the total number of pending requests for the shuttle (i.e., the server) will be possible. It will not be possible to track where the requests are physically waiting. As a result, the actual occupancy of the physical bay areas, especially the bay-in, cannot be monitored. This causes important information about system congestion to be lost. This paper aims to address three essential questions related to this challenge:

1. How do we design, at the preliminary stage, the size of the bays in an AVS/RS to maximize the space efficiency without compromising system performance?
2. Is the previous bay sizing consistently effective? How does it perform during peak production periods? What are the implications of varying arrival rates for storage requests (S-requests) and retrieval requests (R-requests) on bay sizing?
3. What are the implications of alternative server policies on bay sizing? How do these policies influence the probability of vehicle-blocking and waiting time at the bay-in?

The original contribution of this study concerns the modelling of S/R requests as two parallel M/M/1 queues with different finite capacities (N, M) that share a single server, a shuttle. This multi queue single-server model aims to measure the AVS/RS performance in terms of the number of S-requests waiting in the bay-in of a tier and the probability of vehicle-blocking events occurring. The proposed queueing model uses balance equations from a continuous-time Markov chain (CTMC) and assumes exponential service times. However, we show that the performance results we seek are insensitive to the service-time distribution. As such, the proposed model approximates any AVS/RS with general service times. Testing various queue capacities and monitoring their effects on the vehicle-blocking probability and waiting times can guide practitioners to an accurate bay sizing, addressing the research question (1). Finally, a case-study application illustrates how other comparative and competitive S/R server policies affect the system performance under different arrival and service rates, addressing the research questions (2) and (3).

The remainder of this paper is organised as follows: Section 2 presents a literature review of AVS/RSs. Section 3 illustrates the modelling approach, the main elements of the proposed queuing model, and alternative server policies. Section 4 presents and applies the proposed model to a real-world food and beverage industry case study, providing managerial insights. Finally, Section 5 presents the conclusions and directions for further research.

2. Literature review

Previous studies have explored various issues related to AVS/RS and SBS/RS problems. The foundation of this research can be traced back to Malmborg (2002), who introduced an analytical framework for evaluating the performance of AVS/RS. Over time, researchers have predominantly relied on analytical or simulation-based approaches to estimate system performance. Analytical techniques can be classified into travel-time (Lerher et al., 2015; Lerher, 2016; Manzini et al., 2016; Marolt et al., 2022, 2023; Kosanić et al., 2024) or queuing models. In this review, we focus specifically on queuing models. Table 1 summarizes the relevant literature on AVS/RS and SBS/RS, classifying studies based on their key system features:

- System configuration (*System*, Table 1). Two alternative vehicle configurations are adopted in the AVS/RSs and SBS/RSs: tier-to-tier (TT) and tier-captive (TC). In the former, the shuttle can travel among tiers, and no bays are required at the tiers. Only two buffer areas are outside the storage system. Consequently, the *BayCheck* feature in Table 1 is vacant in studies that embrace a TT configuration (Fukunari and Malmborg, 2009; Ekren et al., 2013; Cai, Heragu and Liu, 2014; Ekren et al., 2014). In the TC configuration, shuttles and satellites move within the tier, and lifts provide vertical movement. In these systems, bay-in and bay-out are pivotal to ensure parallel movement and flexibility. This configuration is investigated in this study.
- Modelling approach (*Modelling*, Table 1). Three queuing networks are adopted for modelling AVS/RSs and SBS/RSs: open (Zhang et al., 2009; Marchet et al., 2012; Wang, Mou, and Wu, 2016; Epp, Wiedemann, and Kai, 2017; Ekren and Akpunar 2021; Lupi et al., 2023) closed (Fukunari and Malmborg, 2009), and semi-open queuing networks (Ekren et al., 2013; Cai et al., 2014; Ekren et al., 2014; Roy

Table 1
Literature studies on AVS/RSs and SBS/RSs. Abbreviations: QM: Queuing Model, TC: tier-captive, TT: tier-to-tier, OQN: Open Queuing Network, SOQN: Semi Open Queuing Network, JFQN: Join-Fork Queuing Network, P: Poisson, G: Generic, E: Exponential, Ga: Gaussian, D: deterministic, SD: single-deep, MD: multi-deep, ST: single-tier, MT: multi-tier, MA: multi-aisle, K: finite capacity, ∞: infinite capacity, R: retrieval, S: storage, S/R: storage and retrieval, TH: throughput, CT: cycle time, U: vehicle utilization, VB: vehicle blocking probability, BF: Bay filling degree, E: energy consumption, EQ: external queue at lift-in, TQ(SR): tier queue at shuttle coupling S/R, TQ(S-R): tier queue at shuttle dividing S/R.

Author	System	Modelling	Arrival	Service	Tiers	Bay	S/R	Goal	BayCheck
Fukunari and Malmborg (2009)	TT AVS/RS	CQN	P	E	MT	K	S/R	U	–
Zhang et al. (2009)	TC AVS/RS	OQN	G	G	MT	∞	S/R	CT	No
Marchet et al. (2012)	TC SBS/RS	OQN	P	G	ST	∞	R	CT	No
Ekren et al. (2013)	TT AVS/RS	SOQN	P	E	MT	∞	S/R	TH	–
Cai et al. (2014)	TT AVS/RS	SOQN	G	G	MT	∞	S/R	TH	–
Ekren et al. (2014)	TT AVS/RS	SOQN	P	E	MT,MA	∞	S/R	TH	–
Roy et al. (2015)	TT AVS/RS	SOQN	P	D	MT	∞	S/R	CT,U	–
Zuo et al. (2016)	TC AVS/RS	JFQN	P	G	ST	K	R	CT,U	TQ(SR)
Wang et al. (2016)	TC AVS/RS	OQN	P	Ga	ST	∞	S/R	CT	No
Epp et al. (2017)	TC AVS/RS	OQN	G	G	MT	∞	S/R	CT,U	EQ
Eder (2019)	TC SBS/RS	SOQN	P	G	ST	K	R or S	TH	No
Tappia et al. (2017)	TC AVS/RS	SOQN	P	E	MT	∞	S/R	TH, CT	No
Eder (2020)	TC SBS/RS	SOQN	P	G	ST	K	R or S	TH	No
Ekren and Akpunar (2021)	TC SBS/RS	OQN	G	G	MT	∞	S/R	CT,U,E	TQ(SR)
Eder (2022)	TC SBS/RS	SOQN	P	G	ST	K	R or S	TH	No
Eder (2023a)	TC SBS/RS	SOQN	P	G	ST	K	S/R	CT,TH	No
Eder (2023b)	TC SBS/RS	SOQN	P	E	MA	K	R	TH	No
Lupi et al. (2023)	TC AVSRS	OQN	G	G	MT	K	S/R	CT	TQ(SR)
This study	TC AVS/RS	OQN	P	E	ST	K	S/R	BF,VB	TQ(S-R)

et al., 2015; Eder 2019; Tappia et al., 2017; Eder, 2020, 2022, 2023a).***

- Number of tiers (*Tiers*, Table 1). System performance is investigated by monitoring a single tier (single-tier, ST) under the assumption of a uniform distribution of S/R requests among all tiers or the entire system (multi-tier, MT).
- Arrival and service process (*Arrival, Service*, Table 1). Some fundamental assumptions regarding queuing models consider arrival and service processes. The most commonly used method is the Markovian process. This assumption states that arrivals to the system follow a Poisson distribution, and service times follow an exponential distribution (Poisson, P; Exponential, E). Other studies have presented queuing models with generic distributions to model a wide range of arrival and service time patterns (General, G).
- Goal (*Goal*, Table 1) and type of requests processed (*S/R*, Table 1). Most studies on AVS/RS aim to estimate the S/R cycle time (CT) and system throughput (TH), typically monitoring both S/R requests. Few studies have investigated S- or R-requests individually (Marchet et al., 2012; Zou et al., 2016; Eder, 2020, 2022, 2023a).
- Bay capacity (*Bay*, Table 1). The number of ULs that can be hosted simultaneously determines the bay's capacity. Literature studies mostly assumed infinite queues (∞), neglecting the chance that the bay could become a bottleneck. Recent studies have focused on queuing models with finite capacity (K).
- Bay monitoring (*BayCheck*, Table 1). A finite capacity should be introduced, coupled with a performance indicator that monitors the filling degree of the bays and the probability of having no more space. These performances highlight the role of the bay in the system performance estimation. Most studies ignore this implication. Some have investigated an external queue outside the lift system (External queue, EQ). A few supervised the queues at the tiers (Tier queue, TQ). In the latter case, these studies coupled S/R requests in one queue (TQ(SR)) to estimate the average number of requests waiting for the shuttle/lift rather than the filling degree of the bays. This study aims to fill this gap by dividing S/R requests into two queues (TQ(S-R)).

In this section, we discuss the contributions of previous researchers on the topic of bay capacity, highlighting whether they considered finite or infinite capacities for the bays and whether they treated storage and retrieval requests separately or merged them into a single queue.

Initial, studies on shuttle-based storage systems (SBS/RSs) adopted infinite-capacity queueing models (∞ in *Bay*, Table 1) to represent the list of requests waiting to be handled by shuttles and/or lifts, making it less applicable to real-world warehouse constraints. Zhang et al. (2009) presented a cycle time model for estimating the mean and variance of the S/R cycle time for AVS/RSs, pointing out the relevance of waiting times. Marchet et al. (2012) developed a queueing model to estimate S/R cycle time in SBS/RSs, using an open queueing system with infinite capacity to evaluate shuttle and lift travel times and lift waiting times. Wang, Mou, and Wu (2016) proposed a two-stage open-queuing network (OQN) model in which shuttles and a lift are regarded as servers at different stages to analyse the system performance regarding shuttle waiting time and lift idle periods. Epp et al. (2017) modelled the AVS/RS as a discrete-time OQN to measure R-request time, the number of S-requests in front of the lift (EQ in *BayCheck*, Table 1), and the inter-departure time of R-requests. Tappia et al. (2017) introduced a multi-class semi-open queuing network (SOQN) model to estimate the performance of both single-tier and multi-tier SBS/RSs. Ekren and Akpunar (2021) presented an OQN-based software tool to estimate S/R cycle time, vehicle utilisation, energy consumption, and the number of requests waiting for the shuttle/lift for a predefined SBS/RS warehouse design.

Subsequent works overcame this assumption by modelling the bays and buffers of AVS/RSs and SBS/RSs as queues with finite capacities (K in *Bay*, Table 1). Zuo et al. (2016) presented a model of the AVS/RS as a join-fork queuing network (JFQN), focusing exclusively on retrieval (R) requests. Their analysis only considers the expected waiting time of R-requests in the external buffer and vehicle utilization. They set the buffer capacity to a single request and linked the statuses of bay-in and bay-out to the shuttle's availability to perform S/R requests alternately. This approach, however, limits the system's ability to capture the complexities of both storage and retrieval processes simultaneously, as it did not account for S-requests or consider other buffer capacities beyond a single request. Eder (2019) modeled a Tier-Captive SBS/RS as a continuous-time, discrete open queuing system with limited capacity, aiming to estimate the maximum throughput under specific conditions. Later, the models presented by Eder (2020) and Eder (2022) extended their previous work by introducing changes such as increased storage depth and class-based storage policy. Finally, Eder (2023a) presented a time-continuous open queuing system with a limited capacity to model a TC SBS/RS, where shuttles could serve more aisles in the same tier (aisle-changing). Eder (2023b) expanded their previous work to investigate a multiple-aisle configuration by presenting a queuing-based model based on different queuing systems with a limited capacity to measure throughput in a multiple-aisle TC SBS/RS. However, these models still focused on handling storage or retrieval requests separately. When they were considered together, they were merged into a single queue without distinguishing between the two types of requests. Finally, Lupi et al. (2023) proposed an application-oriented approach to solve an OQN for a TC AVS/RS. In a web-based simulation environment, a data-driven Java modelling tool (JMT) was used to support system performance estimation by modeling lifts, shuttles, satellites, and finite capacity bays. Two classes of jobs (S and R jobs) distinguished between S/R requests to measure the specific impact on the cycle time. Nevertheless, as in the previous studies, both requests were joined in the same queue under the FIFO policy.

In conclusion, the capacity of bays is a critical factor that has been overlooked in the previous queuing models. It has rarely been considered a system bottleneck. Studies usually identify lifting as a bottleneck (Carlo and Vis, 2012; Marchet et al., 2012; Marchet et al., 2013). Though, the lift performance is primarily influenced by the requests sequencing (Zhao et al., 2018) and number of storage levels (Battarra et al., 2022b). Recent studies have shown that adopting double-capacity lifts (Ekren et al., 2023) arranged in systems with a few levels (Battarra et al., 2022b) turns the bay into a system bottleneck. Configurations such as this are widespread in modern AVS/RSs for the food and beverage industry, paving the way for introducing bay capacity as leverage in performance measurements. In addition, this challenge has already been addressed in other storage systems, such as the automated storage and retrieval system (AS/RS). Bozer and Cho (2005) proposed an analytical model for modelling a stacker crane in a multilevel AS/RS. They modelled S/R requests as two parallel queues with infinite capacities, different arrival and service rates, and different set-up times.

Prior studies failed to capture crucial information about the actual occupancy of the bays, by focusing on the overall performance metrics and merging storage and retrieval requests into a single queue. This paper takes a different approach by introducing a novel solution that divides storage and retrieval requests into two queues, with finite and different capacities. The aim is still to monitor key performance metrics such as throughput, waiting times, and system cycle times, but with a specific focus on exploring the impact of the finite capacity of bays on these parameters. This is achieved by testing different server policies and proposing parametric models with finite bay capacities.

3. Methodology

This section discusses the proposed modelling approach and related assumptions (Subsection 3.1), steady-state equations obtained for performance estimation (Subsection 3.2), and the simulation model adopted for validation (Subsection 3.3). Finally, an insensitivity test for the service-time distribution is conducted (Subsection 3.4).

3.1. System model

The critical elements of this warehousing system are material-handling vehicles (lifts, shuttles, and satellites) and bays (bay-ins and bay-outs at the tiers), which play pivotal roles in managing the flow of S/R requests. In the proposed modeling approach, vehicles can be represented as servers and bays as queues. Fig. 2 shows the resulting queuing network.

Initially, we model this system as two parallel birth and death queues sharing a server. S-requests arrive at the storage system with an average interarrival time of $1/\lambda'_S$, where λ'_S is the number of S-requests per unit of time. They wait for the lift-in service in the buffer-in, represented as a queue named Q_{IN}. Upon reaching the destination tier, they wait in the bay-in, modeled as a queue named Q_S with a finite capacity equal to the bay-in capacity. The arrival rate of S-requests at the tier is λ_S. There, they wait for the shuttle service as the R-requests occur at the tier with an average interarrival time of $1/\lambda_R$, where λ_R is the number of R-requests per unit of time. Unlike the S-requests, the R-requests await service in their storage locations within the lane. This "electronic" waiting list is modelled through a queue named Q_R with a finite capacity equivalent to the memory size allocated to store R-requests in the computer memory (Lee, 1997). After service completion, the S-requests leave the system, while the R-requests wait for the lift-out service in bay-out, modelled as a queue named Q_{OUT}, before exiting the system.

Owing to the uniform distribution of S/R requests among all tiers, the system performance can be evaluated by modelling a single tier (Epp et al., 2017; Marchet et al., 2012). In particular, this study focuses on modelling the queues of S/R requests asking for the shuttle at one generic tier (see the multi-queue single-server model in Fig. 2). The proposed analytical approach is based on the following steps.

1. Determining the interarrival time of S/R requests at the storage tier ($1/\lambda_s$ and $1/\lambda_R$, respectively) and the lift-in service time.
2. Determining the shuttle service time ($T_{ServiceS}$, $T_{ServiceR}$).
3. Determining the system performance through a multi-queue single-server model based on two parallel M/M/1 queues with finite capacities N and M served under alternative server policies.

Table 2 presents the main notations used in this section.

The first element (step 1) determines the S/R interarrival times at different tiers. As R-requests are equally distributed among all tiers and occur at the tier, the average interarrival time of R-requests at a generic tier ($T_{ArriveR}$) can be defined as:

$$T_{ArriveR} = nTier\lambda_R \quad (1)$$

S-requests are equally distributed among all tiers; however, the lift-in delays their arrival. The average service time of the lift (T_{LiftIN}) can be determined in several ways: (1) through primary data, as proposed by Battarra et al. (2022b), (2) using analytical formulas, as shown by Eder (2022), or (3) by simulations. In this study, approach (1) is adopted. Given the layout configuration, vehicle kinematics, and storage policy, this analytical–simulation data-driven model provides the lift, shuttle, and satellite service times required to perform an S- or R-request. It

Table 2
Notation of the proposed analytical approach.

Parameter	Description
$nTier$	Number of tiers of the AVS/RS
λ'_S	Arrival rate at the system of S-requests [requests per s]
λ_S	Arrival rate at the tier of S-requests [requests per s]
λ_R	Arrival rate at the tier of R-requests [requests per s]
$T_{ArriveS}$	Inter-arrival time at the tier of S-requests [s]
$T_{ArriveR}$	Inter-arrival time at the tier of R-requests [s]
μ_S	Shuttle service rate for S-requests [service per s]
μ_R	Shuttle service rate for S-requests [service per s]
$T_{ServiceS}$	Average service time at the tier for S-requests [s]
$T_{ServiceR}$	Average service time at the tier for R-requests [s]
T_{LiftIN}	Average service time of lift-in [s]
Q_S	Storage queue for shuttle service
N	Maximum capacity of Q_S
i	Number of S-requests in Q_S simultaneously, i = 0,1,…, N
Q_R	Retrieval queue for shuttle service
M	Maximum capacity of Q_R
j	Number of R-requests in Q_R simultaneously, j = 0,1,…, M
n	Number of requests in the system simultaneously, n = 0,…, M + N + 1
s	Shuttle state: {0: Idle, 1: Processing a S-request, −1: Processing a R-request}
$state(i,j,s)$	The system state is described by three elements: the shuttle state s, the number of waiting requests i in the Q_S, and the number of R-requests j in Q_R
$P_{(i,j,s)}$	Probability of being in state (i,j,s)
P_n	Probability of having n generic requests in the system (0 ≤ n ≤ N + M + 1)
$Pblock_S$	Probability that Q_S is full, i.e., with N S-requests
$Pblock_R$	Probability that Q_R is full, i.e., with M R-requests
$\|S\|$	Total number of Sample States

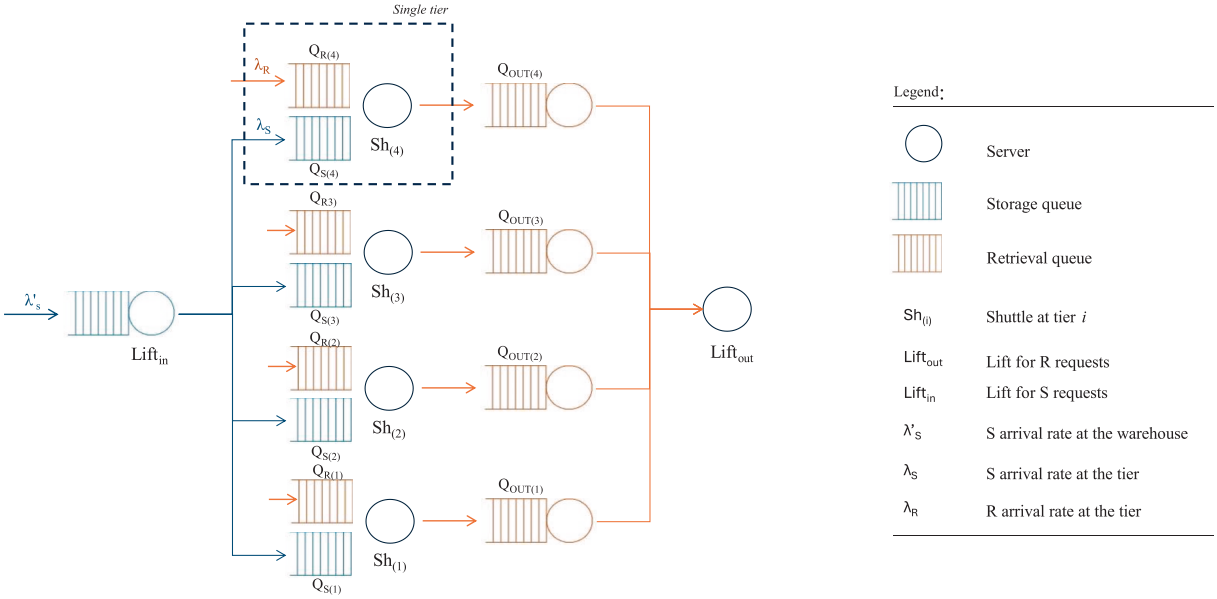


Fig. 2. Queuing network describing a four-tier AVS/RS.

performs an arbitrary number of S/R requests to estimate the average values with a specific confidence interval. The lift-in service time is measured by collecting the time required by the lift-in to load the UL, move it to the destination tier, unload it onto the conveyor, and return it to the ground tier. The travel time is determined by the vehicle velocities, acceleration, deceleration, and fixed times required to load and unload ULs from/to the lift. Given this delay, the average interarrival time of S-requests at a generic tier ($T_{ArriveS}$) can be obtained as:

$$T_{ArriveS} = \frac{\frac{1}{\lambda_S}}{nTier} + T_{LiftIN} \tag{2}$$

The second element (step 2) determines the shuttle service time for performing a S- or R-request ($T_{ServiceS}$ and $T_{ServiceR}$). As for the lift-in service time, the analytical–simulation model is adopted for service-time estimation. The service time for an S-request is the time required by the shuttle to move from the point of completion of the previous request to the bay-in, load the UL, move to the destination lane, release the satellite into the lane to unload the UL at the storage location, and collect the satellite again. Similarly, a R-request involves the time required by the shuttle to move from the point of completion of the last request to the lane where the UL is stored, release the satellite to pick up the UL, collect the satellite again, move to the bay-out, and unload the UL. The travelling times are measured by considering the shuttle and satellite velocities, acceleration, deceleration, and fixed times required to load/unload the ULs from/to the bays and drop/collect the ULs within the lanes. The original contributions provided by the adoption of this model are twofold. First, it facilitates the measurement of the average service times by considering tailored layout configurations and vehicle kinematics. Second, it adopts a heuristic approach to perform S/R requests following the shortest path and combines them according to the shuttle–satellite loading capacities. Moreover, it outperforms the single-cycle (SC) or double-cycle (DC) approaches because the shuttle waits for the subsequent request at the completion point of the previous request.

The third and final element (step 3) is the multi-queue single-server model. This model consists of two parallel M/M/1 queues with finite capacities (N, M) and a shared server because S- and R-requests contend for the same resource: the shuttle. S-requests arrive according to a Poisson arrival process with rate λ_S and wait for the server in the queue Q_S. This queue has a finite capacity N linked to the physical capacity of the bay-in. Similarly, R-requests arrive according to a Poisson arrival process with a rate λ_R and wait for the server in queue Q_R. This queue is an electronic or virtual queue with a finite capacity, M, which represents the list of ULs already within the system that wait at their assigned storage locations until the shuttle becomes available. We can assume a sufficiently large value of M to approximate this queue capacity without constraining the system performance. The requests join the queues Q_S or Q_R under a FIFO policy. Both S/R requests have exponentially distributed service requirements with rates μ_S and μ_R, respectively. The server processes requests according to a specific server policy. This study investigates three different policies, illustrated in Fig. 3 and described as follows:

- Switch policy (*Switch,* Fig. 3). The server processes the requests alternately. An S-request is completed after an R-request. An exception is made if no requests of the other type wait on a server. In this case, the server continues to process the same type of request. The server is idle when no requests are waiting and is free. This policy was introduced to balance the processing of both S- and R-requests and is particularly useful in systems where the arrival rates fluctuate or are not predictable.
- Storage Priority Policy (*SPriority,* Fig. 3). The server prioritizes and processes the S-requests in the queue. It executes an R-request only if no S-requests are waiting in the queue. This policy was introduced to prioritize storage operations, ensuring that ULs were efficiently stored in the system to maintain high throughput for storage operations. It is particularly relevant in systems with high storage demands or where the system's design prioritizes the accommodation of incoming ULs to prevent system congestion.
- Retrieval Priority Policy (*RPriority,* Fig. 3). This is the opposite of the previous policy. In this case, the server prioritizes R-requests. This policy is beneficial in environments where the timely retrieval of stored items is essential, such as in distribution centers or e-commerce fulfillment centers, where customer orders must be retrieved as quickly as possible.

The proposed model uses a continuous-time Markov chain (CTMC) to describe the transitions between different states in the queuing system. The Markovian assumption means that the future state of the system depends only on its current state, not on the sequence of past events. This simplification allows for efficient computation of steady-state probabilities and helps estimate blocking probabilities for storage and retrieval requests. Fig. 4 presents the state transition diagram for the proposed Markov chain model, illustrating how storage (S) and retrieval (R) requests transition between different system states under the Switch policy.

The state diagram consists of nodes and directed edges. Each node represents a system state, while the directed edges show possible transitions based on request arrivals and service completions. The state, $(i, j, s) \in S$ with S denoting the state space, describes the number of S requests (i, with $0 \le i \le N$) waiting in Q_S, number of R-requests (j, with $0 \le j \le M$) waiting in Q_R, and shuttle's status (s). The shuttle can be idle ($s = 0$) or processing a request: $s = 1$ (processing S-request), $s = -1$

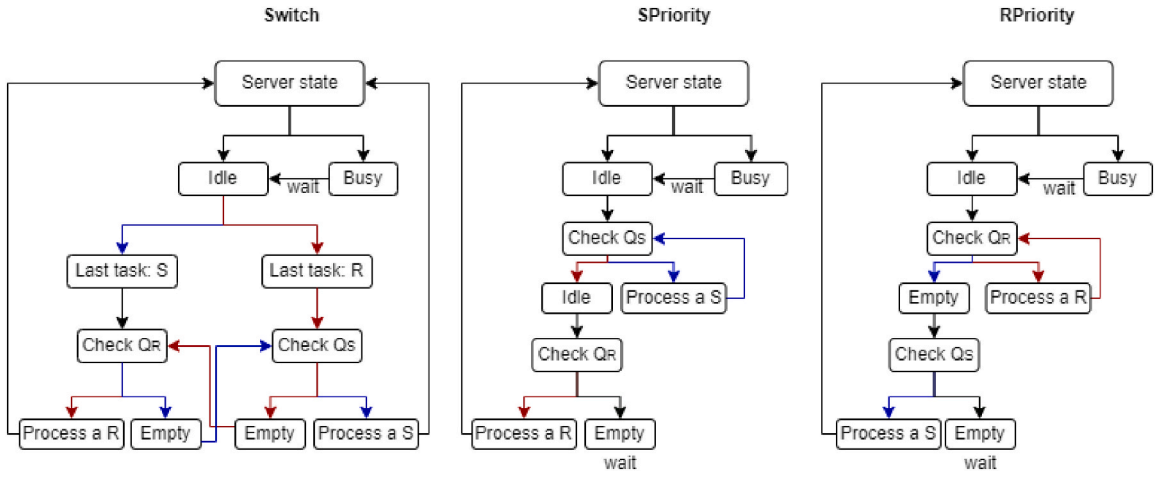


Fig. 3. Server policies: Switch, SPriority, RPriority.

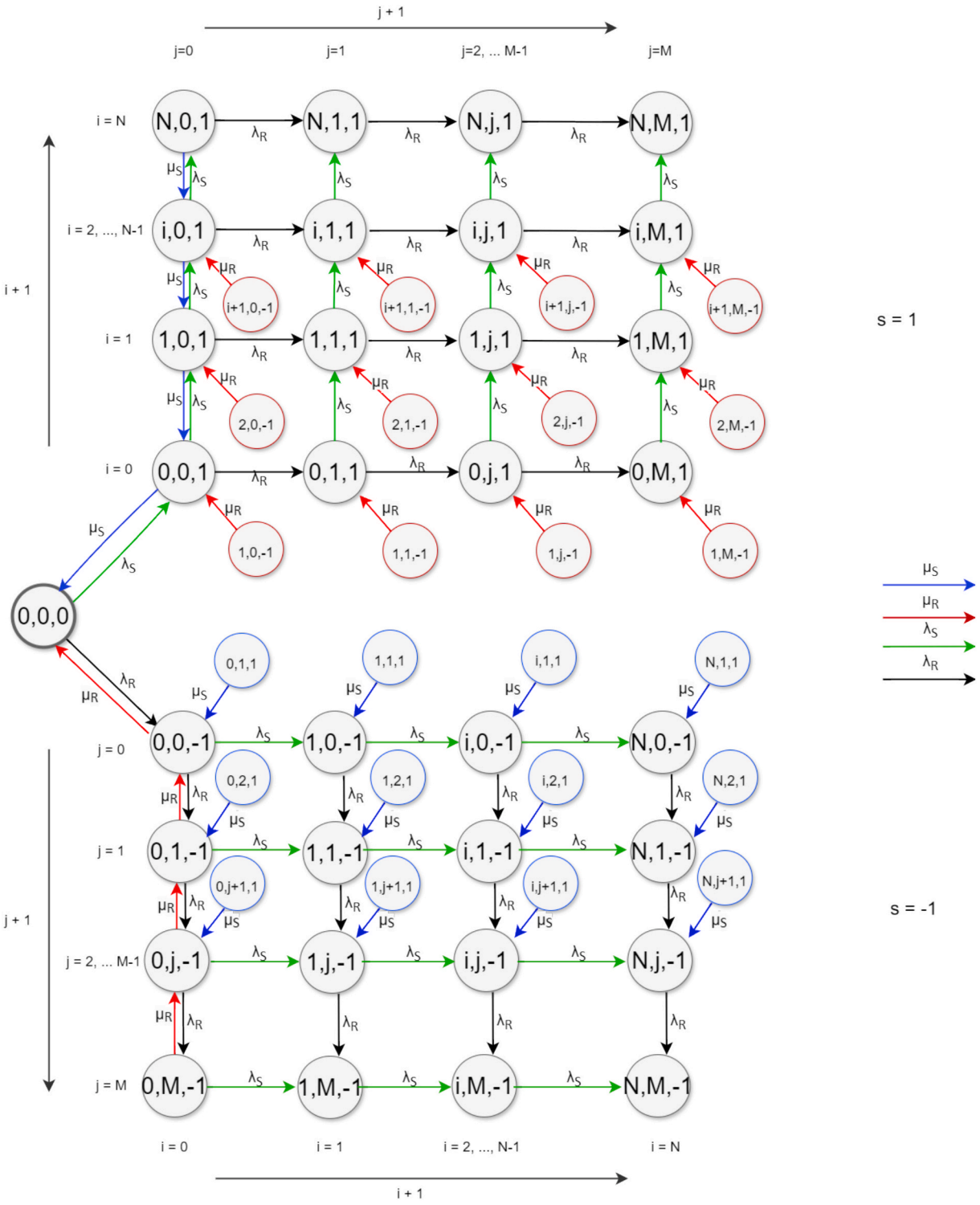


Fig. 4. State space of two M/M/1 queues with finite capacities (N, M) and shared server, operating in Switch Policy.

(processing R-request). The initial state is (0,0,0) and represents the idle condition, during which the server is free and no S/R requests are waiting in the queues.

The edges represent the transitions between states, triggered by different events. The green arrows correspond to S-arrivals (storage requests), black arrows to R-arrivals (retrieval requests), blue arrows indicate completion of the S-service, and red arrows represent completion of the R-service. These transitions are governed by the arrival and service rates, which define how the system evolves.

The state diagram is structured into two symmetrical branches, each corresponding to the type of request the shuttle is processing. The upper branch in Fig. 4 shows the states when the shuttle handles an S-request, represented by $s = 1$, while the lower branch illustrates the states when the shuttle handles an R-request, with $s = -1$. Within each branch, the horizontal and vertical edges represent new S- or R-request arrivals, respectively. The green arrows indicate S-arrivals and black arrows show R-arrivals. The diagonal edges represent the completion of service for these requests, with blue arrows indicating the completion of an S-

request and red arrows representing the completion of an R-request.

The generic state (i, j, 1) represents a condition where the shuttle is busy and processing an S-request ($s = 1$). After service completion, according to the Switch policy, the shuttle moves to the state (i, j-1, −1) if R-requests are waiting in the queue (i.e., if $j > 0$). Otherwise, it processes the S-requests and moves to state (i-1, j, 1). The first column of states (i.e., the ones with $j = 0$ for $s = 1$, and $i = 0$ for $s = -1$ in Fig. 4) shows this particular case, where the shuttle is processing an S- or R-request and only requests of the same type arrive and wait in the queue. By contrast, the rows at the top and bottom of Fig. 4 show the states during which the storage and retrieval queues are full (i.e., $i = N$ or $j = M$). The top and bottom rows in the diagram represent scenarios where either the storage queue (Q_S) or retrieval queue (Q_R) reaches its full capacity, causing blocking events. In these cases, new arrivals (storage or retrieval requests) cannot be processed until space is available in the respective queue.

Regardless of the server policy, the number of states for S-requests can increase to N as up to M R-requests can be collected in the computer memory. Let $|S|$ denote the total number of states, defined as:

$$|S| = [(N+1)(M+1)]2+1 \tag{3}$$

In this equation, $N + 1$ represents the possible states of queue Q_S (from 1 to N), including the initial state where no elements are waiting (0). $M + 1$ represents the same, but for Q_R. The server can be occupied by an S- or R-request; therefore, this sum must be multiplied by two. Finally, the initial state (0,0,0) is included.

In the long term, the arrival rates of S/R requests must be balanced (i.e., $\lambda_S = \lambda_R$). However, the proposed analytical model distinguishes between the two rates because some systems can show a marked difference from one period to another, induced by seasonal phenomena during the year or week. For instance, the food-and-beverage-industry production lines are fully operational on weekdays, whereas they slow down or stop on weekends.

3.2. Stationary distribution

Steady-state balance equations, or balance equations, provide the probability of the system being in each state once the Markov chain reaches the steady state. To clarify, these equations indicate likely the system will be in any given state after it has stabilized over time. These equations are adapted to account for the finite capacities of the queues, represented by N (max number of S-requests in the storage queue) and M (max number of R-requests in the retrieval queue).

Equation (4) represents the starting condition, where no requests are pending. Equations (5)–(13) cover all states where the server is handling a storage request (states with $s = 1$ in Fig. 4). Similarly, Equations (14)–(22) describe the system when the server is handling a retrieval request (states with $s = -1$ in Fig. 4). Finally, Equation (23) represents the conservation of probability. It ensures that the total probability across all possible states is normalized to 1.

Solving these equations produces the probabilities of the system being in each possible state at steady state. Appendix A contains the detailed steady-state equations for the different priority policies: see Equations (35)–(54) for the Storage Priority policy and Equations 55–74 for the Retrieval Priority policy.

$$P_{0,0,0}\,(\lambda_S + \lambda_R) = P_{0,0,1}\,\mu_S + P_{0,0,-1}\,\mu_R \tag{4}$$

$$P_{0,0,1}(\lambda_S + \lambda_R + \mu_S) = P_{1,0,1}\mu_S + P_{0,0,0}\lambda_S + P_{1,0,-1}\mu_R \tag{5}$$

$$P_{i,0,1}(\lambda_S + \lambda_R + \mu_S) = P_{i+1,0,1}\,\mu_S + P_{i-1,0,1}\,\lambda_S + P_{i+1,0,-1}\mu_R,\ \ i = 1,\ \ldots,\ N-1 \tag{6}$$

$$P_{N,0,1}(\lambda_R + \mu_S) = P_{N-1,0,1}\ \lambda_S\ , i = N \tag{7}$$

$$P_{0,j,1}(\lambda_S + \lambda_R + \mu_S) = P_{0j-1,1}\ \lambda_R + P_{1,j,-1}\mu_R, j = 1, \cdots, M-1 \tag{8}$$

$$P_{0,M,1}(\lambda_S + \mu_S) = P_{0,M-1,1}\ \lambda_R + P_{1,M,-1}\,\mu_R, j = M \tag{9}$$

$$P_{i,j,1}(\lambda_S + \lambda_R + \mu_S) = P_{i-1,j,1}\,\lambda_S + P_{i,j-1,1}\,\lambda_R + P_{i+1,j,-1}\,\mu_R,\ \ i = 1, \ldots,\ N-1;\ j\ =\ 1,\ \ldots,\ M-1 \tag{10}$$

$$P_{N,j,1}(\lambda_R + \mu_S) = P_{N-1,j,1}\lambda_S + P_{Nj-1,1}\lambda_R, j = 1, \ldots, M-1 \tag{11}$$

$$P_{i,M,1}(\lambda_S + \mu_S) = P_{i-1,M,1}\lambda_S + P_{i,M-1,1}\lambda_R + P_{i+1,M,-1}\mu_R,\ i\ =\ 1,\ \ldots,\ N-1 \tag{12}$$

$$P_{N,M,1}(\mu_S) = P_{N-1,M,1}\lambda_S + P_{N,M-1,1}\lambda_R \tag{13}$$

$$P_{0,0,-1}(\lambda_S + \lambda_R + \mu_R) = P_{0,1,-1}\mu_R + P_{0,0,0}\lambda_R + P_{0,1,1}\mu_S \tag{14}$$

$$P_{0,j,-1}(\lambda_S + \lambda_R + \mu_R) = P_{0,j+1,-1}\mu_R + P_{0,j-1,-1}\lambda_R + P_{0,j+1,1}\mu_S,\ \ j = 1,\ \ldots,\ M-1 \tag{15}$$

$$P_{0,M,-1}(\lambda_S + \mu_R) = P_{0,M-1,-1}\lambda_R \tag{16}$$

$$P_{i,0,-1}(\lambda_S + \lambda_R + \mu_R) = P_{i-1,0,-1}\lambda_S + P_{i,1,1}\mu_S,\quad i = 1, \ldots, N-1 \tag{17}$$

$$P_{N,0,-1}(\lambda_R + \mu_R) = P_{N-1,0,-1}\lambda_S + P_{N,1,1}\mu_S \tag{18}$$

$$P_{i,j,-1}(\lambda_S + \lambda_R + \mu_R) = P_{i-1,j,-1}\lambda_S + P_{i,j-1,-1}\lambda_R + P_{i,j+1,1}\mu_S,\ i = 1, \ldots, N-1; j = 1, \ldots, M-1 \tag{19}$$

$$P_{N,j,-1}(\lambda_R + \mu_R) = P_{N-1,j,-1}\lambda_S + P_{N,j-1,-1}\lambda_R + P_{N,j+1,1}\mu_S,\ \ j = \ 1,\ \ldots,\ M-1 \tag{20}$$

$$P_{i,M,-1}(\lambda_S + \mu_R) = P_{i-1,M,-1}\lambda_S + P_{i,M-1,-1}\lambda_R, i = 1, \ldots, N-1 \tag{21}$$

$$P_{N,M,-1}(\mu_R) = P_{N-1,M,-1}\lambda_S + P_{N,M-1,-1}\lambda_R \tag{22}$$

$$\sum_{i=0}^{N}\sum_{j=0}^{M}\sum_{s=-1}^{1} P_{i,j,s} = 1 \tag{23}$$

These probabilities provide the foundation for evaluating key performance metrics of the AVS/RS system. The number of S/R requests in the system ($L_{SR}, L_S L_R$) reflects the overall load. The system times (T_S, T_R) account for both waiting and service times. The blocking probabilities ($Pblock_S, Pblock_R$) for the storage and retrieval queues show when the queues are full and unable to accept new requests. Finally, shuttle utilization (U) measures how frequently the shuttle is in use.

- Average number of S/R requests in the system (L_{SR})

$$L_{SR} = \sum_{i=0}^{N}\sum_{j=0}^{M}\sum_{s=-1}^{1}(i+j+|s|\,)P_{(i,j,s)} \tag{24}$$

This is the weighted sum of the probability of being in the state (i,j,s) and the number of S/R requests in the system in that state. It is zero when the server is idle, and no requests are waiting in the queues.

- Average number of S-requests in the system (L_S)

$$L_S = \sum_{i,j,s:i>0\ \text{or}\ s=1}(i+s)P_{(i,\,j,\,s)} \tag{25}$$

This is the weighted sum of the probability of being in the state (i,j,s) and the number of S-requests in the system in that state.

- Average number of R-requests in the system (L_R)

$$L_R = \sum_{i,j,s:j>0\ \text{o}\ s=-1}(j+|s|\,) \tag{26}$$

This is the weighted sum of the probability of being in the state (i,j,s) and the number of R-requests in the system in that state.

- The average system (waiting plus service) time for an S-request (T_S) can be approximated as

$$T_S = \frac{L_S + L_R}{2\lambda_S} \tag{27}$$

The expected storage time is when an S-request travels through the system from entry (i.e., when the request occurs at bay-in) to exit (i.e., when the UL is stored within the lane), including both the waiting time in the queue and service time.

- The average system time for an R-request (T_R) can be approximated as

$$T_R = \frac{L_S + L_R}{2\lambda_R} \tag{28}$$

The expected retrieval time is the time required for an R-request to travel through the system from entry (when the request occurs) to exit (when the UL reaches the bay-out). This includes both the waiting time in the queue and service time.

- Number of S-requests in the storage queue (L_{Q_s})

$$L_{Q_S} = \sum_{i,j,s:i>0} iP_{(i,j,s)} \tag{29}$$

This value represents the average number of S-requests in queue Q_S. It ranges from 0 to N, which is the queue capacity. This differs from L_S because it ignores the S-requests onboard the shuttle (being processed).

- Number of R-requests in the retrieval queue (L_{Q_R})

$$L_{Q_R} = \sum_{i,j,s:j>0} jP_{(i,j,s)} \tag{30}$$

This value represents the average number of R-requests in queue Q_R. It ranges from 0 to M, which is the queue capacity. This differs from L_R because it ignores the R-requests onboard the shuttle (being processed).

- Probability of Blocking in Q_S ($Pblock_S$)

$$P_{\text{block}S} = \sum_{N,j,s} P_{(i,j,s)} \tag{31}$$

The term blocking probability indicates that the entire system is filled and lacks space for more ULs to enter. $Pblock_S$ describes the likelihood of a full bay-in. This is the sum of the probabilities of being in the state (i,j,s) where i=N.

- Probability of Blocking in Q_R ($Pblock_R$)

$$P_{\text{block}R} = \sum_{i,M,s} P_{(i,j,s)} \tag{32}$$

$Pblock_R$ represents the probability of the electronic queue Q_R being full. This is the sum of the probabilities of being in the states (i,j,s) where j = M.

- Vehicle utilization (U)

$$U = 1 - P_0 \tag{33}$$

Shuttle utilisation is measured as the complementary probability of the emptiness of the queuing system (P_0 with $n = 0$). The server must wait because there are no S- or R-requests in the queuing system. This corresponds to the state (0,0,0).

3.3. Model validation

A Python-based simulation model was developed to test the accuracy of the proposed analytical model. The simulation model ran the arriving S/R requests according to the Poisson arrival processes and with exponential service times. We assumed a Switch server policy and FIFO queue policy in the simulations. Regarding the AVS/RS layout, we modelled a tier-captive deep-lane AVS/RS inspired by the one proposed in the case-study application. It comprised nine tiers connected by two single-capacity lifts dedicated to S/R requests. The rack has a double-sided cross aisle with deep lanes on both sides. There are 24 lanes on each side; each lane can host up to 20 ULs. The shuttle and the satellite handle one UL at a time. The bay-in and bay-out at the tiers have finite capacities equal to three ULs. In the generic tier, an S-request arrives on average every 53.9 min, whereas an R-request arrives every 47.6 min. The shuttle takes, on average, 1.92 min and 2.11 min to perform S/R requests, respectively.

The simulation model was run for 36 replications with 37,730 S/R requests processed per replication. Table 3 shows the steady-state performance values monitored in each replication. The number of simulation replications was determined based on a convergence criterion that monitored the stability of the output performance metrics, i.e., by observing the expected time gap (see columns *Gap(S)* and *Gap(R)* in Table 3). These values showed the difference between the expected time and obtained time. The expected time was measured as the average of times of the previous replications (see columns T_S and T_R in Table 3), while the obtained time includes the current replication in the mean. The difference decreased as the number of replications increased. The replications 34, 35, and 36 had a zero gap. Finally, the last three rows in the table report the average performance values obtained from 36 simulation replications, the values generated by the proposed analytical model, and the gaps between them. The results clearly show that the analytical model performs reasonably well under the switch policy, with a maximum gap of 1 %.

Table 3
Simulation model performance. Comparison with the analytical model.

Replication	T_S [s]	T_R [s]	L_{SR}	U [%]	Gap(S) [%]	Gap(R) [%]
1	121.51	131.07	0.072	6.97	/	/
2	120.41	129.05	0.072	6.92	−0.90	−1.54
3	120.93	130.01	0.072	6.98	−0.01	−0.01
4	122.23	130.33	0.073	6.99	0.26	0.05
5	117.63	131.94	0.072	6.94	−0.60	0.28
6	120.11	131.61	0.072	6.99	−0.06	0.14
7	120.19	131.29	0.072	7.00	−0.03	0.07
8	120.42	133.98	0.073	7.05	0.00	0.31
9	118.45	130.08	0.071	6.93	−0.18	−0.09
10	124.53	130.03	0.073	7.05	0.36	−0.08
11	121.37	130.19	0.072	6.99	0.05	−0.05
12	121.75	130.64	0.073	7.01	0.07	−0.01
13	120.88	132.19	0.073	7.02	0.01	0.08
14	123.34	131.95	0.073	7.07	0.15	0.05
15	120.79	131.44	0.073	7.00	−0.01	0.02
16	120.49	134.34	0.073	7.08	−0.02	0.16
17	121.60	129.77	0.072	7.00	0.03	−0.07
18	120.32	129.01	0.072	6.92	−0.03	−0.09
19	121.56	129.20	0.072	6.96	0.03	−0.07
20	119.49	131.48	0.072	6.96	−0.06	0.02
21	120.59	131.33	0.072	6.99	−0.01	0.01
22	120.38	129.51	0.072	6.95	0.00	−0.05
23	124.17	130.10	0.073	7.05	−0.01	−0.03
24	121.42	132.05	0.073	7.04	0.00	0.04
25	118.50	129.57	0.071	6.89	−0.06	−0.04
26	122.20	131.94	0.073	7.06	−0.01	0.03
27	120.59	130.55	0.072	6.98	0.00	−0.01
28	121.01	130.31	0.072	6.99	0.00	−0.02
29	120.62	131.47	0.072	7.00	0.00	0.02
30	118.70	131.03	0.072	6.93	0.00	0.00
31	120.22	129.72	0.072	6.95	0.00	−0.03
32	120.89	130.73	0.072	6.98	0.00	0.00
33	120.34	130.02	0.072	6.97	0.00	−0.02
34	120.84	130.85	0.072	6.99	0.00	0.00
35	120.82	130.84	0.072	6.98	0.00	0.00
36	120.83	130.89	0.072	6.95	0.00	0.00
Average	120.84	130.85	0.072	6.99		
Analytical	122.05	131.20	0.073	6.99		
Gap	1.01	0.27	0.94	0.02		

3.4. Insensitivity to service time distribution

Finally, the insensitivity to service-time distribution was tested. This property implies that the mean system performance metrics depend only on the mean service time and not the service-time distribution. This study used the abovementioned simulation to test three distributions: exponential, uniform, and normal. It was run for 36 replications with 37,730 S/R requests processed for each time distribution. The comparison was based on the average expected time required to process an S- or R-request. The uniform distribution generated expected times, on an average, that were 1,61 % lower than that obtained through exponential distributions, whereas the normal distribution took the gap to −0.88 %. All observed values are reported in Table 8, Appendix B.

Given an average expected time of a couple of minutes for an S- or R-request, it is possible to state that this gap turns into a difference of a few seconds difference. For instance, Battarra et al. (2022b) studied a real-world AVS/RS. Adopting a data-driven approach, they monitored an average expected time of 120 s for processing an S- or R-request. This resulted in gaps of 1.97 and 1.08 s, respectively.

In conclusion, it can be argued that the proposed analytical model is relatively insensitive to service-time distribution. Thus, any service time distribution can be used in the analysis by equating the service rate to the reciprocal of the mean service time to obtain all mean performance metrics.

4. Case study

This section presents and applies the proposed analytical model to a deep-lane tier-captive AVS/RS from the food and beverage industry. Given the capacity of the actual bays, it aims to measure and monitor the bay-in filling degree (i.e., Q_S length) and the probability of having a full bay-in under the previously introduced server policies: Switch, SPriority, and RPriority. A multiscenario analysis assesses how the system performance varies under different server policies and S/R arrival rates. In conclusion, Subsection 4.1 explores managerial insights on how business managers can effectively utilize the proposed model for designing and managing real-world AVS/RSs across different industrial contexts.

The proposed AVS/RS consists of nine independent tiers connected by two lifts: one for S- and one for R-requests. The lifts handle the requests in a single cycle (SC) under a FIFO policy, one at a time, with the capacity of one. Once each service is completed, it returns to the ground tier. The bay-in and bay-out at the tiers are adjacent to the related lifts. They can host up to three ULs simultaneously. A conveyor system is the interface between the lifts and bays. One shuttle and one satellite with single-loading capacity serve the requests at each system tier (tier-captive configuration). There are 34 double-sided lanes per tier (x-direction) that can host up to 10 ULs each (z-direction). A mono SKU-lane storage policy is adopted, ensuring that only items of the same SKU are stored within a lane. Table 4 lists the features of the AVS/RS layouts.

The period analyzed covers one month. S-requests occur every 110 s, whereas R-requests occur every 87 s. Assuming a uniform distribution of requests among all tiers, the arrival rates at the generic tier become 1 S-request every 462 s and 1 R-request every 410 s. Concerning the service time, the average service time for performing an S-request is 115 s, whereas a R-request takes 126 s. These service times are determined using the previously mentioned hybrid model, which requires the layout parameters (see Table 4) and vehicle kinematics (see Table 5) as inputs. Vehicle kinematics are obtained from real-world AVS/RS associated with the food-and-beverage industry.

Finally, the maximum Q_S capacity is set to 3 according to the bay-in capacity ($N = 3$), whereas the maximum Q_R capacity is not limited by any physical constraint. This value has been fixed to 3 to maintain a balanced system and monitor the frequency of three or more R-requests waiting in the queue ($M = 3$). In addition, maintaining a low M ensures high service levels, such as fast delivery.

Fig. 5 shows the resulting state diagrams for the three server policies. The total number of states is 25, of which 12 are associated with a shuttle processing an S-request and 12 with an R-request. The edges connecting the states vary according to server policy. The switch policy (see Fig. 5a) has edges moving from the bottom to the top branch and vice versa because the S/R requests are processed alternately. The SPriority policy (see Fig. 5b) has edges extending from the bottom to the top to prioritize S-requests. Conversely, the RPriority policy (see Fig. 5c) has edges that move from the top branch to the bottom branch to prioritize R-requests.

Table 6 presents the system performance under three different scenarios, all of which assume a bay capacity of 3 ULs. The SPriority policy stands out for offering the lowest expected processing times for both S- and R-requests (see T_S and T_R in Table 6). However, this advantage comes at the cost of a higher blocking percentage in queue Q_R(see $Pblock_R$). On the contrary, the RPriority policy strikes a balance between the other two policies in terms of expected processing times; however, it also exhibits a relatively high blocking percentage in queue Q_S(see $Pblock_S$). The Switch policy, as the name suggests, offers a compromise between the two, with blocking probabilities in both Q_S and Q_R falling in between those of the SPriority and RPriority policies.

This comparison highlights how each policy performs under different conditions and provides valuable insights into selecting an optimal policy based on specific objectives. For example, if fast delivery is a key competitive advantage for the company, RPriority policy might be the best choice. This is because it simultaneously minimizes the expected time for processing R-requests (T_R) and reduces the probability of having three or more ULs waiting for processing in queue Q_R(see $Pblock_R$).

The proposed multi-queue single-server model offers a twofold advantage. In existing warehouses, it supports the testing of alternative server policies during the steady-state phase. These policies can prioritize different performance aspects, thus helping to address the research question (3). On the contrary, in green-field cases, during the preliminary design phase of an AVS/RS, the model acts as a valuable tool. It allows testing different bay capacities and server policies to identify the

Table 4
AVS/RS features. Case study.

Parameter	Number	High (y) [m]	Width (x) [m]	Length (z) [m]
Tiers	9	2.25	37.68	38
Sides	2 (left, right)	2.25	37.68	38
Lanes	34 lanes per side	2.25	1.57	1.90
Storage Locations	10 ULs per lane	2.25	1.57	1.90
Bays	18 (9 bay-in, 9 bay-out)	/	1.57	5.7
Buffers	2 (1 buffer-in, 1 buffer-out)	/	1.57	/

Table 5
Vehicle kinematics for shuttle, satellite, and lift. Case study.

Parameter	Description	Value
vLift, $vLift_e$	Maximum lift speed with/without load	1 [m/s]
vConv	Maximum conveyor speed	0.30 [m/s]
vSh, vSh_e	Maximum shuttle speed with/without load	3 [m/s]
vSat, $vSat_e$	Maximum satellite speed with/without load	1 / 1.2 [m/s]
aLift, $aLift_e$	Lift acceleration/deceleration with/without load	0.35 [m/s^2]
aSh, aSh_e	Shuttle acceleration/deceleration with/without load	0.8 / 1.2 [m/s^2]
aSat, $aSat_e$	Satellite acceleration/deceleration with/without load	1 / 1.2 [m/s^2]
tLoad, tUnload	Fixed UL load/unload time on/from the shuttle or lift	1.5 [s]
tCollect, tDrop	Fixed UL collect/drop time on/from the satellite	6.5 / 7.5 [s]
tCoupling	Fixed coupling/uncoupling time of satellite on/from the shuttle	5 [s]

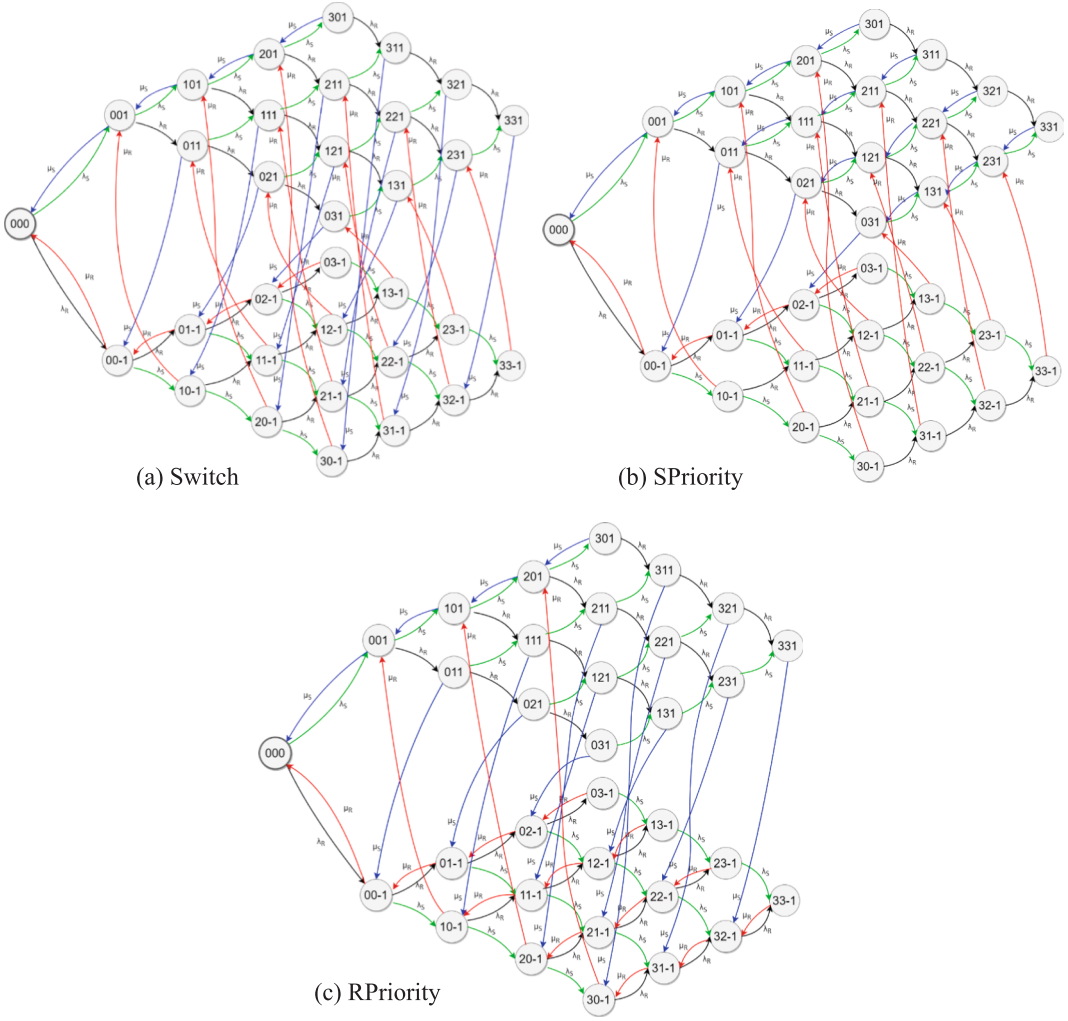


Fig. 5. Server Policies in a double-queue single-server model with finite capacities.

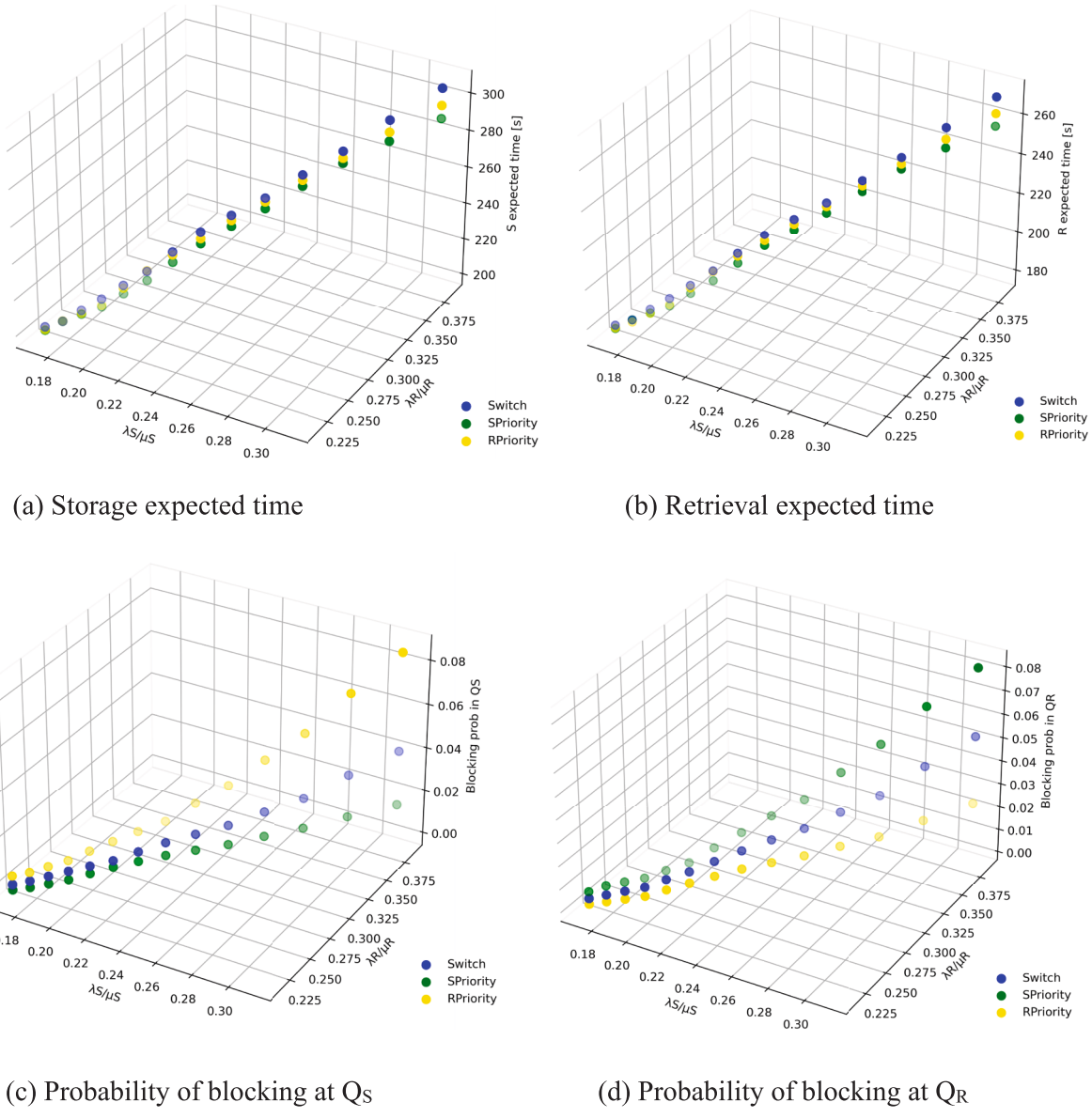
Table 6
System performance under alternative server policies. Case study.

Performance	Description	Switch	Spriority	Rpriority
L_{SR}	Number of S/R requests in the system	1.12	1.09	1.11
L_S	Number of S-requests in the system	0.50	0.43	0.58
L_R	Number of R-requests in the system	0.61	0.66	0.52
T_S [s]	Expected time for S-request	257.51	251.73	255.43
T_R [s]	Expected time for R-request	228.37	223.25	226.53
$1/\lambda_S$	S Inter-arrival time	462.32	462.32	462.32
$1/\lambda_R$	R Inter-arrival time	410.00	410.00	410.00
$1/\mu_S$	S-request Service time	115.50	115.50	115.50
$1/\mu_R$	R-request Service time	126.81	126.81	126.81
$Pblock_S$ [%]	Vehicle-blocking probability in Q_S	1.62	0.72	3.50
$Pblock_R$ [%]	Vehicle-blocking probability in Q_R	2.18	3.53	1.01
U [%]	Vehicle utilization	52.16	52.06	52.14
λ_S / μ_S	Ratio arrival/service per S-requests	0.2498	0.2498	0.2498
λ_R / μ_R	Ratio arrival/service per R-requests	0.3093	0.3093	0.3093

best compromise between the space reserved for bays and vehicle-blocking probability. Doing so helps answer research question (1). Increasing the number of ULs that a bay can host (i.e., its capacity) leads to a lower probability of vehicle-blocking; however, it also reduces the storage space.

To address the research question (2), the proposed model is tested under several arrival rates to monitor how the results are influenced by this factor. A multiscenario analysis under various server policies and arrival rates is conducted. The resulting 3D plot diagrams are presented in Fig. 6.

The plot diagrams depicted in Fig. 6(a, b) present the average expected times for S- and R-requests. Fig. 6(c, d) present the probabilities of having three or more requests waiting in Q_S and Q_R. They provide more detailed insight into the effect of the arrival rate on selecting the optimal policy. Specifically, Fig. 6(b) shows that the SPriority policy is the best for minimizing the expected time of S-requests, as long as the ratio between λ_S and μ_S remains above 0.20. When the ratio decreases from 0.20 to 0.18, the performance of the SPriority and RPriority policies becomes similar. Hence, the gap between the two average expected times is lower than 1 s. Finally, for ratios lower than 0.18, all three policies become equivalent. Similarly, the SPriority policy exhibits the best performance when the ratio between λ_R and μ_R is greater than 0.25. The performance becomes comparable to that of the Switch policy for ratios between 0.25 and 0.22, and to the performances of all policies become equivalent when the ratio drops below 0.22. All observed values are reported in Table 7, Appendix B. Similar inferences can be drawn



(a) Storage expected time

(b) Retrieval expected time

(c) Probability of blocking at Q_S

(d) Probability of blocking at Q_R

Fig. 6. System performance visualization: 3D plot diagrams. Case study.

from the plot diagrams in Fig. 6(c, d) representing the vehicle-blocking probabilities in Q_S and Q_R. In general, ensuring the likelihood of being in a state where one or both bays are full allows the storage system to work without delays and at maximum efficiency.

Given the arrival and service rates of an existing AVS/RS or the expected ones for a new AVS/RS, these diagrams provide a valuable tool for researchers to study the impact of alternative server policies on the S/R expected times and filling degree of the bay-in. The previously mentioned results showed that all policies performed similarly when the arrival rates were low (i.e., when the interarrival times were high). In these conditions, the likelihood of having an S- or R-request already waiting in the queue when a new request occurred was almost zero. In this case, prioritizing S- or R-requests or processing them alternately did not affect the allocation of the server to the requests. In contrast, when the system was stressed (i.e., under a considerable workload) and the interarrival time was short, selecting optimal policies could lead to significantly different system performances. The maximum gaps observed in this analysis were around 6 % for the expected time between the SPriority and Switch policies, and 7 % for the blocking probability between the SPriority and RPriority policies. These gaps occurred at a ratio of 0.31 for λ_S and μ_S, and 0.39 for λ_R and μ_R, as shown in Fig. 6.

In conclusion, this multiscenario analysis should be replicated by practitioners, for different arrival and service rates of tailored configurations of AVS/RSs to measure the system performance and test different bay capacities (when possible). The following subsection exploits two key insights on how business managers can effectively utilize the proposed model for designing and managing real-world AVS/RSs across different industrial contexts.

4.1. Managerial insights

This section presents two key managerial insights derived from the proposed queuing model, focusing on its practical applications for optimizing the performance of AVS/RSs. By analyzing the system performance and exploring different layout configurations, warehouse managers can make more informed decisions regarding the designs of these storage systems. These insights are valuable for new system designs and the operational enhancement of existing systems.

The first insight relies on the general distribution of vehicle service time. The results from the case-study application demonstrate the

versatility and practicality of Markov chains in modelling complex queuing systems such as the multi queue single-server configuration used for the AVS/RS. A key strength of the proposed approach lies in its insensitivity to the specific distributions of service times, making it a robust tool for various real-world applications. Thus, any service-time distribution can be used in the analysis by equating the service rate to the reciprocal of the mean service time in the balance equations, i.e., $\mu = 1/E$ [service time]. This flexibility is particularly valuable in systems where the service-time distribution varies across different industrial contexts, such as for the AVS/RS. The three most relevant elements that affect this distribution are the following:

1 The shuttle and satellite kinematics (e.g., speed with or without load, acceleration, deceleration) and their quantity. The vehicle's kinematic primary depends on the type of unit load handled. Full palletised unit loads are bulkier and heavier than smaller plastic totes, requiring lower speed and more braking space and time. Moreover, in deep-lane AVS/RSs the presence of one or more satellites impacts service times in a twofold way. Firstly, decoupling from the shuttle and operating in parallel can speed up storage and retrieval operations. This is called a free-satellite configuration. Secondly, increasing the number of satellites per shuttle enhances service times, even if there is no straightforward linear correlation between an increase in satellites and a decrease in service times, as higher numbers may lead to congestion. Battarra et al. (2022b) have thoroughly explored these aspects.
2 The storage policy. It plays a pivotal role in affecting service time distribution, as how items are stored within the warehouse impacts the operational efficiency of the shuttle and satellite. Literature studies usually adopt random storage policies. Nevertheless, real-world AVS/RSs usually perform S/R requests according to the mono SKU-lane storage policy. In these cases, no relocation cycles are required.
3 The lane depth. It can significantly influence service times because longer lanes require satellites to travel greater distances. This effect is particularly pronounced between units accessing the nearest position to the main aisle and those accessing the farthest position, leading to substantial variations in service times and potentially causing higher congestion and delays. The (2) and (3) elements have been thoroughly explored by Lupi et al. (2024).

The second insight explores how the proposed model provides a valuable decision-making tool that supports the design and size of input bays for new or existing AVS/RS, helping managers make informed decisions to enhance the overall system performance. Achieving the best trade-off between bay sizing and system performance is pivotal in designing an AVS/RS. Dedicating more space to input bays reduces the risk of congestion and system blocking, particularly under high arrival rates, thereby improving the throughput and responsiveness. However, increasing the bay capacity involves trade-offs: it reduces the space available for storage, potentially lowering the system's storage density. From a managerial perspective, the results of this study can guide the identification of an optimal bay capacity that minimizes the risk of queuing delays and system idle times without incurring unnecessary space-related costs.

The results provided by the multiscenario analysis in the case-study application (see plot diagrams in Fig. 6) show how managers can simulate and monitor system performances under alternative server policies and by adjusting the S/R arrival rates or bay-in capacity parameters. This flexibility provides crucial insights into identifying potential bottlenecks and optimizing resource allocation. Furthermore, the ability to model the filling degree of the input bays enhances the system's predictive power, allowing for more accurate estimations of the system throughput and probability of vehicle-blocking events under actual operative conditions and prospects of growth. This information is essential for ensuring that bays are adequately sized to handle peak loads efficiently.

5. Conclusion and further research

This study introduced an original analytical model for a tier-captive deep-lane AVS/RS to monitor the number of S-requests waiting in bay-in at the tier and the probability of vehicle-blocking events. The aim was to provide a helpful decision-support tool for bay sizing in the warehouse preliminary design phase and system-performance monitoring under given bay capacities, varying arrival rates, and server policies. The proposed OQN was described as a continuous-time Markov Chain. The state diagram helped describe all possible states and measure the steady-state probabilities of being in each state. The novel contributions of the proposed analytical model are as follows:

- Dividing the S/R requests into two parallel queues Q_S and Q_R, with different finite capacities (N, M). This multi-queue single-server model has two objectives. First, the two-parallel queue approach enables monitoring of the bay-in's filling degree, where only S-requests wait to be processed. Second, introducing different finite capacities enables discerning the small finite capacity of the bay-in, constrained by the rack configuration, from the electronic queue of R-requests with no physical constraints.
- Introducing different arrival rates for S/R requests (i.e., λ_S and λ_R). It aims to support system performance estimation under specific periods when the system shows a marked difference between S and R arrivals, such as during the peak of shipping (outbound activities) or production (incoming activities).
- Introducing and comparing different server policies is an alternative to the traditional FIFO policy adopted in the literature.

These three novel contributions support the use of the proposed analytical model in a twofold manner. First, for existing warehouses (1), during the regime phase, the proposed model facilitates testing of alternative server policies and arrival rates that prioritize one performance over another. For instance, in some cases, reducing the probability of vehicle blocking at the retrieval queue could be more relevant than that at the storage queue, to ensure fast shipping. Second, for new warehouses (2), during the preliminary design phase, the proposed analytical model could be a valuable tool for testing different bay capacities, server policies, and arrival rates to identify the best compromise between the amount of space to be reserved for the bays and system performance. Space efficiency is pivotal in AVS/RS performance, and space is finite. The user-friendly nature of this model provides added value for practitioners.

The proposed model was applied to real-world AVS/RSs from the food-and-beverage industry to demonstrate its practical application. The results indicated that the probability of vehicle blocking in Q_S was not irrelevant and that it varied according to the server policy adopted, proving that distinguishing S/R requests into two queues was mandatory for monitoring this probability and estimating the filling degree of the bay-in. Moreover, a multiscenario analysis compared three alternative server policies and various arrival rates. The resulting plot diagrams provided helpful insights for better scheduling of incoming and outgoing requests within the storage system and throughout time and layout improvements. Future research investigations that capitalize on the results of this study are expected to:

- Adding bay-out to the proposed queuing model. Given a single-tier view, bay-out can be represented as a single queue with a single server, i.e., lift-out, because it collects the R requests picked up by the shuttle.
- Exploring the effect of an increase (or decrease) in the bay-in capacity on the system performance. A multiscenario analysis should be conducted on the relationship between various arrival rates and bay capacities to monitor their impacts on the system performance.

This analysis supports defining the maximum capacity required by bay-in and bay-out to satisfy ULs' incoming and outgoing requests irrespective of the arrival rate.

- Introducing storage-capacity utilization as the state diagram's fourth element. Storage capacity refers to the overall number of storage locations within the AVS/RS, which is the product of the number of tiers, lanes, and storage locations per lane. The arrival or completion of a request modifies the storage capacity utilization. Monitoring the effects of alternative arrival and service rates and server policies can ensure that the system never explodes or enters a stock-out.
- Extending the model to represent multiple shuttles per tier. This would require a reformulation of the queuing structure from M/M/1 to M/M/n, where n is the number of resources available at each tier. Such an extension would facilitate system performance evaluation in larger-scale AVS/RS configurations and represent a relevant direction for future scalability analysis.

CRediT authorship contribution statement

Ilaria Battarra: Writing – original draft, Data curation, Validation, Conceptualization, Formal analysis. **Melike Baykal-G Rsoyü:** Writing – review & editing, Methodology, Conceptualization, Supervision. **Riccardo Manzini:** Conceptualization, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

$$P_{0,0,0}(\lambda_S+\lambda_R)=P_{0,0,1}\mu_S+P_{0,0,-1}\mu_R \tag{34}$$

$$P_{0,0,1}(\lambda_S+\lambda_R+\mu_S)=P_{1,0,1}\mu_S+P_{0,0,0}\lambda_S+P_{1,0,-1}\mu_R \tag{35}$$

$$P_{i,0,1}(\lambda_S+\lambda_R+\mu_S)=P_{i+1,0,1}\mu_S+P_{i-1,0,1}\lambda_S+P_{i+1,0,-1}\mu_R,\ \mathrm{i}=1,\ldots,\mathrm{N}-1 \tag{36}$$

$$P_{N,0,1}(\lambda_R+\mu_S)=P_{N-1,0,1}\lambda_S \tag{37}$$

$$P_{0,j,1}(\lambda_S+\lambda_R+\mu_S)=P_{0,j-1,1}\lambda_R+P_{1,j,-1}\mu_R+P_{1,j,1}\mu_S,\ j=1,\ldots,M-1 \tag{38}$$

$$P_{0,M,1}(\lambda_S+\mu_S)=P_{0,M-1,1}\lambda_R+P_{1,M,-1}\mu_R+P_{1,M,1}\mu_S \tag{39}$$

$$P_{i,j,1}(\lambda_S+\lambda_R+\mu_S)=P_{i-1,j,1}\lambda_S+P_{i,j-1,1}\lambda_R+P_{i+1,j,-1}\mu_R+P_{i+1,j,1}\mu_S,\quad i=1,\ldots,N-1;j=1,\ldots,M-1 \tag{40}$$

$$P_{N,j,1}(\lambda_R+\mu_S)=P_{N-1,j,1}\lambda_S+P_{N,j-1,1}\lambda_R,\quad j=1,\ldots,M-1 \tag{41}$$

$$P_{i,M,1}(\lambda_S+\mu_S)=P_{i-1,M,1}\lambda_S+P_{i,M-1,1}\lambda_R+P_{i+1,M,-1}\mu_R+P_{i+1,M,1}\mu_S,\ \mathrm{j}\ =\ 1,\ \ldots,\ \mathrm{N}-1 \tag{42}$$

$$P_{N,M,1}(\mu_S)=P_{N-1,M,1}\lambda_S+P_{N,M-1,1}\lambda_R \tag{43}$$

$$P_{0,0,-1}(\lambda_S+\lambda_R+\mu_R)=P_{0,1,-1}\mu_R+P_{0,0,0}\lambda_R+P_{0,1,1}\mu_S \tag{44}$$

$$P_{0,j,-1}(\lambda_S+\lambda_R+\mu_R)=P_{0,j+1,-1}\mu_R+P_{0,j-1,-1}\lambda_R+P_{0,j+1,1}\mu_S,\ j\ =\ 1,\ \ldots,\ M-1 \tag{45}$$

$$P_{0,M,-1}(\lambda_S+\mu_R)=P_{0,M-1,-1}\lambda_R \tag{46}$$

$$P_{i,0,-1}(\lambda_S+\lambda_R+\mu_R)=P_{i-1,0,-1}\lambda_S, i=1,\ldots,N-1 \tag{47}$$

$$P_{N,0,-1}(\lambda_R+\mu_R)=P_{N-1,0,-1}\lambda_S \tag{48}$$

$$P_{i,j,-1}(\lambda_S+\lambda_R+\mu_R)=P_{i-1,j,-1}\lambda_S+P_{i,j-1,-1}\lambda_R,\ i=1,\ldots,N-1;j=1,\ldots,M-1 \tag{49}$$

$$P_{N,j,-1}(\lambda_R+\mu_R)=P_{N-1,j,-1}\lambda_S+P_{N,j-1,-1}\lambda_R, j=1,\ldots,M-1 \tag{50}$$

$$P_{i,M,-1}(\lambda_S+\mu_R)=P_{i-1,M,-1}\lambda_S+P_{i,M-1,-1}\lambda_R, i=1,\ldots,N-1 \tag{51}$$

$$P_{N,M,-1}(\mu_R)=P_{N-1,M,-1}\lambda_S+P_{N,M-1,-1}\lambda_R \tag{52}$$

$$\sum_{i,j,s}P_{i,j,s}=1 \tag{53}$$

Steady-state balance equations under the Retrieval Priority policy:

$$P_{0,0,0}(\lambda_S+\lambda_R)=P_{0,0,1}\mu_S+P_{0,0,-1}\mu_R \tag{54}$$

$$P_{0,0,1}(\lambda_S+\lambda_R+\mu_S)=P_{1,0,1}\mu_S+P_{0,0,0}\lambda_S+P_{1,0,-1}\mu_R \tag{55}$$

$$P_{i,0,1}(\lambda_S+\lambda_R+\mu_S)=P_{i+1,0,1}\mu_S+P_{i-1,0,1}\lambda_S+P_{i+1,0,-1}\mu_R,\ i=1,\ldots,N-1 \tag{56}$$

$$P_{N,0,1}(\lambda_R+\mu_S)=P_{N-1,0,1}\lambda_S \tag{57}$$

$$P_{0,j,1}(\lambda_S+\lambda_R+\mu_S)=P_{0,j-1,1}\lambda_R, j=1,...,M-1 \quad (58)$$

$$P_{0,M,1}(\lambda_S+\mu_S)=P_{0,M-1,1}\lambda_R \quad (59)$$

$$P_{i,j,1}(\lambda_S+\lambda_R+\mu_S)=P_{i-1\,j,1}\lambda_S+P_{i,j-1,1}\lambda_R+P_{i+1\,j,-1}\mu_R,\ i=1,...,N-1;j=1,...,M-1 \quad (60)$$

$$P_{N,j,1}(\lambda_R+\mu_S)=P_{N-1\,j,1}\lambda_S+P_{N,j-1,1}\lambda_R, j=1,...,M-1 \quad (61)$$

$$P_{i,M,1}(\lambda_S+\mu_S)=P_{i-1,M,1}\lambda_S+P_{i,M-1,1}\lambda_R, i=1,...,N-1 \quad (62)$$

$$P_{N,M,1}(\mu_S)=P_{N-1,M,1}\lambda_S+P_{N,M-1,1}\lambda_R \quad (63)$$

$$P_{0,0,-1}(\lambda_S+\lambda_R+\mu_R)=P_{0,1,-1}\mu_R+P_{0,0,0}\lambda_R+P_{0,1,1}\mu_S \quad (64)$$

$$P_{0,j,-1}(\lambda_S+\lambda_R+\mu_R)=P_{0,j+1,-1}\mu_R+P_{0,j-1,-1}\lambda_R+P_{0,j+1,1}\mu_S,\ j=1,\cdots,M-1 \quad (65)$$

$$P_{0,M,-1}(\lambda_S+\mu_R)=P_{0,M-1,-1}\lambda_R \quad (66)$$

$$P_{i,0,-1}(\lambda_S+\lambda_R+\mu_R)=P_{i-1,0,-1}\lambda_S+,P_{i,1,1}\mu_S+P_{i,1,-1}\mu_R,\ i=1,...,N-1 \quad (67)$$

$$P_{N,0,-1}(\lambda_R+\mu_R)=P_{N-1,0,-1}\lambda_S+P_{N,1,1}\mu_S+P_{N,1,-1}\mu_R \quad (68)$$

$$P_{i,j,-1}(\lambda_S+\lambda_R+\mu_R)=P_{i-1\,j,-1}\lambda_S+P_{i,j-1,-1}\lambda_R+P_{i,j+1,1}\mu_S+P_{i,j+1,-1}\mu_R,\ i=1,...,N-1;j=1,...,M-1 \quad (69)$$

$$P_{N,j,-1}(\lambda_R+\mu_R)=P_{N-1\,j,-1}\lambda_S+P_{N,j-1,-1}\lambda_R+P_{N,j+1,1}\mu_S+P_{N,j+1,-1}\mu_R,\ j=1,...,M-1 \quad (70)$$

$$P_{i,M,-1}(\lambda_S+\mu_R)=P_{i-1,M,-1}\lambda_S+P_{i,M-1,-1}\lambda_R, \mathrm{i}=1,\ \cdots,\ \mathrm{N}-1 \quad (71)$$

$$P_{N,M,-1}(\mu_R)=P_{N-1,M,-1}\lambda_S+P_{N,M-1,-1}\lambda_R \quad (72)$$

$$\sum_{i,j,s}P_{i,j,s}=1 \quad (73)$$

Appendix B

Table 7. System performance comparison under variant arrival rates. Scenario 0% illustrates the arrival rates adopted for S and R requests in the case study applications.

Policy	Variation	−20 %	−15 %	−10 %	−5%	0 %	5 %	10 %	15 %	20 %	25 %	30 %	35 %	40 %	45 %
	λ_S / μ_S	0.31	0.29	0.28	0.26	0.25	0.24	0.23	0.22	0.21	0.20	0.19	0.19	0.18	0.17
	λ_R / μ_R	0.39	0.36	0.34	0.33	0.31	0.29	0.28	0.27	0.26	0.25	0.24	0.23	0.22	0.21
Switch	T_S [s]	304.39	290.86	277.32	267.42	257.51	250.56	243.60	234.80	226.00	220.07	214.14	209.50	204.85	203.16
	T_R [s]	269.94	257.94	245.94	237.16	228.37	222.20	216.03	208.23	200.97	195.17	189.91	185.79	181.67	180.17
	$Pblock_S$ [%]	4.00	3.20	2.40	2.00	1.60	1.40	1.20	0.95	0.70	0.60	0.50	0.40	0.30	0.25
	$Pblock_R$ [%]	5.20	4.20	3.20	2.70	2.20	1.90	1.60	1.30	1.00	0.80	0.60	0.55	0.50	0.45
SPriority	T_S [s]	288.12	279.50	270.87	261.30	251.73	244.49	237.24	229.11	220.97	215.40	209.83	207.34	204.85	201.25
	T_R [s]	255.51	247.87	240.22	231.73	223.23	216.81	210.39	203.18	195.96	191.03	186.09	183.88	181.67	178.48
	$Pblock_S$ [%]	1.50	1.25	1.00	0.85	0.70	0.65	0.60	0.50	0.40	0.25	0.10	0.05	0.00	0.00
	$Pblock_R$ [%]	8.10	6.75	5.40	4.45	3.50	2.95	2.40	1.90	1.40	1.20	1.00	0.95	0.90	0.75
RPriority	T_S [s]	295.15	284.37	273.58	264.51	255.43	247.74	240.04	233.16	226.28	218.52	210.76	207.86	204.96	201.64
	T_R [s]	261.74	252.18	242.62	234.58	226.53	219.70	212.87	206.77	200.67	193.79	186.91	183.72	180.52	178.82
	$Pblock_S$ [%]	8.50	6.95	5.40	4.45	3.50	2.85	2.20	1.90	1.60	1.30	1.00	0.85	0.70	0.65
	$Pblock_R$ [%]	2.30	1.85	1.40	1.20	1.00	0.90	0.80	0.65	0.50	0.35	0.20	0.20	0.20	0.20

Table 8. Performance measures under three different time distributions: Uniform, Exponential and Normal.

	Normal			Uniform			Exponential		
Rep	L_{SR}	T_S [s]	T_R [s]	L_{SR}	T_S [s]	T_R [s]	L_{SR}	T_S [s]	T_R [s]
1	0.071	118.32	129.09	0.071	119.43	132.51	0.072	121.51	131.07
2	0.070	118.40	128.73	0.071	119.68	131.92	0.072	120.41	129.05
3	0.071	118.30	128.99	0.071	119.46	132.27	0.072	120.93	130.01
4	0.071	118.32	129.01	0.072	119.90	132.26	0.073	122.23	130.33
5	0.071	119.13	129.59	0.071	118.11	128.94	0.072	117.63	131.94
6	0.071	117.78	129.83	0.072	118.07	128.64	0.072	120.11	131.61
7	0.071	117.86	129.52	0.072	118.60	129.25	0.072	120.19	131.29
8	0.072	118.08	132.17	0.073	118.18	128.77	0.073	120.42	133.98
9	0.070	116.16	128.33	0.071	116.58	129.81	0.071	118.45	130.08

(continued on next page)

(*continued*)

	Normal			Uniform			Exponential		
Rep	**L_{SR}**	**T_S [s]**	**T_R [s]**	**L_{SR}**	**T_S [s]**	**T_R [s]**	**L_{SR}**	**T_S [s]**	**T_R [s]**
10	0.072	122.12	128.28	0.072	122.56	129.76	0.073	124.53	130.03
11	0.071	119.02	128.44	0.072	119.45	129.92	0.072	121.37	130.19
12	0.071	119.39	128.88	0.072	119.83	130.36	0.073	121.75	130.64
13	0.072	118.54	130.41	0.072	118.97	131.91	0.073	120.88	132.19
14	0.072	120.95	130.17	0.073	121.39	131.67	0.073	123.34	131.95
15	0.071	118.45	129.67	0.072	118.88	131.16	0.073	120.79	131.44
16	0.072	118.15	132.53	0.073	118.58	134.06	0.073	120.49	134.34
17	0.071	119.24	128.02	0.072	119.68	129.49	0.072	121.60	129.77
18	0.070	117.99	127.27	0.071	118.42	128.73	0.072	120.32	129.01
19	0.071	119.20	127.46	0.071	119.64	128.93	0.072	121.56	129.20
20	0.071	117.17	129.71	0.072	117.60	131.20	0.072	119.49	131.48
21	0.071	118.25	129.56	0.072	118.68	131.06	0.072	120.59	131.33
22	0.071	118.05	127.77	0.071	118.48	129.24	0.072	120.38	129.51
23	0.072	121.76	128.34	0.072	122.21	129.82	0.073	124.17	130.10
24	0.072	119.07	130.27	0.072	119.50	131.77	0.073	121.42	132.05
25	0.070	116.20	127.82	0.071	116.62	129.29	0.071	118.50	129.57
26	0.072	119.83	130.16	0.072	120.27	131.66	0.073	122.20	131.94
27	0.071	118.26	128.79	0.072	118.69	130.27	0.072	120.59	130.55
28	0.071	118.66	128.55	0.072	119.10	130.03	0.072	121.01	130.31
29	0.071	118.28	129.70	0.072	118.71	131.19	0.072	120.62	131.47
30	0.071	116.40	129.26	0.071	116.82	130.75	0.072	118.70	131.03
31	0.071	117.89	127.97	0.071	118.32	129.44	0.072	120.22	129.72
32	0.071	118.55	128.97	0.072	118.98	130.46	0.072	120.89	130.73
33	0.071	118.01	128.27	0.071	118.44	129.74	0.072	120.34	130.02
34	0.071	118.49	129.08	0.072	118.93	130.57	0.072	120.84	130.85
35	0.071	118.47	129.08	0.072	118.91	130.56	0.072	120.82	130.84
36	0.071	118.49	129.13	0.072	118.92	130.62	0.072	120.83	130.89
	Average	0.071	118.53	129.13	0.072	119.02	130.50	0.072	120.84
	Gap [%]	−1.64	−1.91	−1.31	−0.90	−1.51	−0.26		

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Ilaria, B., Accorsi, R., Manzini, R., & Rubini, S. (2022a). Storage efficiency in a deep-lane AVS/RS. *IFAC-PapersOnLine, 55*(10), 1337–1342. https://doi.org/10.1016/j.ifacol.2022.09.576

Battarra Ilaria, Riccardo Accorsi, Riccardo Manzini & Sara Rubini. (2022b). Hybrid model for the design of a deep-lane multisatellite AVS/RS. The International Journal of Advanced Manufacturing Technology, (121): 1191–1217, 2022. doi:10.1007/s00170-022-09375-x.

Ilaria, B., Riccardo, A., Giacomo, L., & Riccardo, M. (2024a). A design framework for shuttle-based automated storage systems. *IFACPaperOnline, 58*(19), 1216–1221. https://doi.org/10.1016/j.ifacol.2024.09.086

Battarra Ilaria, Accorsi Riccardo, Lodini Alberto, Lupi Giacomo, Manzini Riccardo, Sirri Gabriele (2024b). Storage space efficiency in deep-lane autonomous vehicle storage and retrieval system. In book: Warehousing and Material Handling Systems for the Digital Industry. Doi: 10.1007/978-3-031-50273-6_14.

Bozer, Y. A., & Cho, M. (2005). Throughput performance of automated storage/retrieval systems under stochastic demand. *IIE Transactions, 37*(4), 367–378. https://doi.org/10.1080/07408170590917002

Xiao, C., Heragu, S. S., & Liu, Y. (2014). Modeling and evaluating the avs/rs with tier-to-tier vehicles using a semi-open queueing network. *IIE Transactions, 46*(9), 905–927. https://doi.org/10.1080/0740817X.2013.849832

Hector, C., & Iris, V. (2012). Sequencing dynamic storage systems with multiple lifts and shuttles. *International Journal of Production Economics, 140*, 844–853. https://doi.org/10.1016/j.ijpe.2012.06.035

Eder, Michael (2019). An analytical approach for a performance calculation of shuttle-based storage and retrieval systems. *Production & Manufacturing Research, 7*(1), 255–270. https://doi.org/10.1080/21693277.2019.1619102

Eder, Michael (2020). An approach for a performance calculation of shuttle-based storage and retrieval systems with multiple-deep storage. *The International Journal of Advanced Manufacturing Technology, 107*, 59–873. https://doi.org/10.1007/s00170-019-04831-7

Eder, Michael (2022). An analytical approach for a performance calculation of shuttle-based storage and retrieval systems with multiple-deep and class-based storage. *Production & Manufacturing Research, 10*(1), 321–336. https://doi.org/10.1080/21693277.2022.2083715

Eder, Michael (2023a). An analytical approach of multiple-aisle shuttle-based storage and retrieval systems. *The International Journal of Advanced Manufacturing Technology, 127*, 1585–1596. https://doi.org/10.1007/s00170-023-11485-z

Eder, Michael (2023b). Analytical approach of merging a different number of storage aisle under a fully sequenced order. *The International Journal of Advanced Manufacturing Technology, 124*, 487–496. https://doi.org/10.1007/s00170-022-10519-2

Ekren, B. Y., & Akpunar, A. (2021). An open queuing network-based tool for performance estimations in a shuttle-based storage and retrieval system. *Applied Mathematical Modelling, 89*(2), 1678–1695. https://doi.org/10.1016/j.apm.2020.07.055

Ekren, B. Y., Heragu, S. S., Krishnamurthy, A., & Malmborg, C. J. (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*, 205–215. https://doi.org/10.1109/TASE.2012.2200676

Ekren, B. Y., Heragu, S. S., Krishnamurthy, A., & Malmborg, C. J. (2014). Matrix-geometric solution for semi-open queuing network model of autonomous vehicle storage and retrieval system. *Computers Industrial Engineering, 68*, 78–86. https://doi.org/10.1016/j.cie.2013.12.002

Ekren, B. Y., Lerher, T., Küçükyaşar, M., & Jerman, B. (2023). Cost and performance comparison of tier-captive SBS/RS with a novel AVS/RS/ML. *International Journal of Production Research, 62*(5), 1648–1662. https://doi.org/10.1080/00207543.2023.2199101

Epp, Martin, Wiedemann, S., & Kai, Furmans (2017). A discrete-time queueing network approach to performance evaluation of autonomous vehicle storage and retrieval systems. *International Journal of Production Research, 55*(4), 960–978. https://doi.org/10.1080/00207543.2016.1208371

Fukunari, Miki, & Malmborg, C. J. (2009). A network queuing approach for evaluate on of performance measures in autonomous vehicle storage and retrieval systems. *European Journal of Operational Research, 193*, 152–167. https://doi.org/10.1016/j.ejor.2007.10.049

Kamali, Ali (2019). Smart warehouse vs. traditional warehouse – review. *CiiT International Journal of Automation and Autonomous System, 11*, 9–16.

Kosanić Nenad, Jakob Marolt, Nenad Zrnić & Tone Lerher. (2024). Travel time model for multiple-deep shuttle-based storage and retrieval systems. International Journal of Production Research. 2024. Vol. 62, No. 7, 2606–2639.

Lee, H. F. (1997). Performance analysis for automated storage and retrieval systems. *IIE Transactions, 29*(1), 15–28. https://doi.org/10.1080/07408179708966308

Lerher, Tone (2016). Travel time model for double-deep shuttle-based storage and retrieval systems. *International Journal of Production Research, 54*(9), 2519–2540. https://doi.org/10.1080/00207543.2015.1061717

Lerher, Tone, Ekren, B. Y., Dukic, G., & Rosi, B. (2015). Travel time model for shuttle-based storage and retrieval systems. *The International Journal of Advanced Manufacturing Technology, 78*, 1705–1725. https://doi.org/10.1007/s00170-014-6726-2

Lupi, Giacomo, Accorsi, R., Battarra, I., & Manzini, R. (2023). Configuration of an AVS/RS using a data-driven queueing network model. *Lecture Notes in Mechanical*

Engineering, Springer, Cham, 381–390. https://doi.org/10.1007/978-3-031-34821-1_42

Lupi, Giacomo, Accorsi, R., Battarra, I., Manzini, R., & Sirri, G. (2024). Space efficiency and throughput performance in AVS/RS under variant lane depths. *The International Journal of Advanced Manufacturing Technology, 131*, 1449–1466. https://doi.org/10.1007/s00170-024-13160-3

Malmborg, C. J. (2002). Conceptualizing tools for autonomous vehicle storage and retrieval systems. *International Journal of Production Research, 40*(8), 1807–1822. https://doi.org/10.1080/00207540110118668

Manzini, Riccardo, Accorsi, R., Baruffaldi, G., Cennerazzo, T., & Gamberi, M. (2016). Travel time models for deep-lane unit-load autonomous vehicle storage and retrieval system (AVS/RS). *International Journal of Production Research, 54*(14), 4286–4304. https://doi.org/10.1080/00207543.2016.1144241

Marchet, Gino, Melacini, M., Perotti, S., & Tappia, E. (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. https://doi.org/10.1080/00207543.2011.639815

Marchet, Gino, Melacini, M., Perotti, S., & Tappia, E. (2013). Development of a framework for the design of autonomous vehicle storage and retrieval system. *International Journal of Production Research, 51*(14), 4365–4387. https://doi.org/10.1080/00207543.2013.778430

Marolt Jackob, Nenad Kosanić & Tone Lerher. (2022). Relocation and storage assignment strategy evaluation in a multiple-deep tier captive automated vehicle storage and retrieval system with undetermined retrieval sequence. The Int ernational Journal of Advanced Manufacturing Technology 118, 3403–3420 (2022). doi:10.1007/s00170-021-08169-x .

Jackob, M., Šinko, S., & Lerher, T. (2023). Model of a multiple-deep automated vehicles storage and retrieval system following the combination of Depth-first storage and Depth-first relocation strategies. *International Journal of Production Research, 61*(15), 4991–5008. https://doi.org/10.1080/00207543.2022.2087568

Debjit, R., Krishnamurthy, A., Heragu, S. S., & Malmborg, C. J. (2015). Stochastic models for unit-load operations in warehouse systems with autonomous vehicles. *Annals of Operations Research, 231*, 129–155. https://doi.org/10.1007/s10479-014-1665-8

Elena, T., Roy, D., De Koster, R., & Melacini, M. (2017). Modeling, analysis, and design insights for shuttle-based compact storage systems. *Transportation Science, 51*(1), 269–295. https://doi.org/10.1287/trsc.2016.0699

Wang, Yanyan, Mou, Shandong, & Wu, Yaohua (2016). Storage assignment optimization in a multi-tier shuttle warehousing system. *Chinese Journal of Mechanical Engineering, 29*(2), 421–429. https://doi.org/10.3901/CJME.2015.1221.152

Zhao Ning, Luo Lei & Lodewijks Gabriel. (2018). Scheduling two lifts on a common rail considering acceleration and deceleration in a shuttle based storage and retrieval system. Computers & Industrial Engineering. Volume 124, 2018, 48-57. ISSN 0360-8352. doi:10.1016/j.cie.2018.07.007.

Zhang, Li, Krishnamurthy, A., Malmborg, C. J., & Heragu, S. S. (2009). Variance-based approximations of transaction waiting times in autonomous vehicle storage and retrieval systems. *European Journal of Industrial Engineering, 3*, 146–169. https://doi.org/10.1504/EJIE.2009.023603

Zou, Bipan, Xianhao, Xu., Gong, Y. Y., & De Koster, R. (2016). Modeling parallel movement of lifts and vehicles in tier-captive vehicle-based warehousing systems. *European Journal of Operational Research, 254*(1), 51–67. https://doi.org/10.1016/j.ejor.2016.03.039