

An analytical performance investigation of RCS/RS under a class-based access structure over the stack height

Philipp Trost & Michael Eder

To cite this article: Philipp Trost & Michael Eder (2025) An analytical performance investigation of RCS/RS under a class-based access structure over the stack height, Production & Manufacturing Research, 13:1, 2530519, DOI: [10.1080/21693277.2025.2530519](https://doi.org/10.1080/21693277.2025.2530519)

To link to this article: <https://doi.org/10.1080/21693277.2025.2530519>



© 2025 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 16 Jul 2025.



Submit your article to this journal [↗](#)



View related articles [↗](#)



View Crossmark data [↗](#)



An analytical performance investigation of RCS/RS under a class-based access structure over the stack height

Philipp Trost and Michael Eder

Institute of Engineering Design, Technische Universität Wien, Vienna, Austria

ABSTRACT

The requirements for modern storage systems are steadily increasing due to limited space, cost, time, and personnel. Robotic compact storage and retrieval systems (RCS/RS), where containers are stacked and arranged in a block layout with robots operating from above, offer a promising solution. Some systems benefit from a self-sorting effect, where robots relocate previously moved containers after accessing non-directly accessible ones, resulting in demand-based sorted stacks. Despite various analytical models for automated storage systems, RCS/RS remain under-researched. Apart from two distinct papers on performance evaluation, there are no general, fast, and easy-to-use tools to assess system throughput under demand-based access patterns. Additionally, the performance benefits of self-sorting have not yet been studied. This paper presents an analytical approach to predict RCS/RS performance using a class-based access structure. A discrete event simulation validates the model, and an optimization example demonstrates the model's broad applicability and ease of use.

ARTICLE HISTORY

Received 2 September 2024

Accepted 19 June 2025

KEYWORDS

Automated storage systems; robotic compact storage and retrieval systems; class-based storage strategy; cycle time model; probability-based demand

1. Introduction

The major challenges facing modern supply chains and the logistics sector today can be summarised as demographic changes and a shortage of labour, the high costs of land and real estate, and the substantial growth in demand driven by the ongoing trend towards e-commerce. The looming threat of climate change, and the consequent need for sustainability, further exacerbates these challenges. Warehousing, with its essential functions of storage and buffering, is invariably associated with high costs due to low productivity.

In this context, automated or fully autonomous compact storage systems, characterised by high storage densities, high potential throughput rates, and high availability, offer solutions to the challenges discussed above.

Robotic compact storage and retrieval systems (RCS/RS) are one example of a warehouse that meets these requirements. Some of these systems employ return

CONTACT Philipp Trost  philipp.trost@tuwien.ac.at  Institute of Engineering Design, Technische Universität Wien, Vienna, Austria

© 2025 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

relocations, meaning that containers previously relocated are immediately return relocated after retrieving the required container. The retrieved container is then placed back into the warehouse on top of any stack. This process results in a warehouse organised from top to bottom by demand – i.e. from low to high demand – following the Last-In-First-Out (LIFO) principle. [Figure 1](#) illustrates the sorting principle used by *AutoStore*, with A-movers located in the upper sections of the stacks and C-movers at the base. [Table 1](#) provides a list of abbreviations used in this study.

A class-based access structure may enable higher throughput rates, particularly when most of the demand is concentrated in the upper range of the container stacks. This results in shorter relocation times, as these occur less frequently. Considering these effects may influence the early stages of the warehouse selection process. Nowadays, material handling providers simulate nearly every new storage system before it is sold to the customer and initiated. An analytical tool to predict the expected performance would be less time-consuming and computationally intensive. Aside from two relevant analytical approaches, there are neither fast nor straightforward methods available.



Figure 1. RCS/RS type *AutoStore* showing the principle of self sorting and ABC slotting (source: Autostore (2023)).

Table 1. Abbreviations.

3D-AS/RS	3-dimensional automatic storage and retrieval systems
AS/RS	Automatic storage and retrieval systems
CTM	Cycle time model
DCC	Dual command cycle
I/O point	Input and output point
LIFO	Last-In-First-Out
MQ-LC	Multi queue with limited capacity
ORCS/RS	Overhead robotic compact storage and retrieval systems
RCS/RS	Robotic compact storage and retrieval systems
RMFS	Robotic mobile fulfilment system
SBS/RS	Shuttle-based storage and retrieval systems
SCC	Single command cycle
SOQN	Semi-open queuing network
SQ-LC	Single queue with limited capacity

Furthermore, none of the existing analytical approaches for RCS/RS takes into account an article demand structure. Consequently, there are no universally valid statements on the performance of RCS/RS under a class-based access distribution. Additionally, there are neither commercial standards nor reliable market research data providing information on expected performance.

Given the research gap described above, this paper aims to present an analytical approach based on a cycle time model developed by Trost and Eder (2024b), considering a class-based zoning over the stack height, based on a *Pareto*-distributed demand structure. Despite the exemplary application of the *Pareto* distribution, any arbitrary access structure can be implemented.

This objective, along with the identified research gap, allows to formulate the following research questions, which this paper seeks to answer:

- (1) How can the cycle time of a single robot operating in an RCS/RS, considering a class-based storage policy, be determined analytically?
- (2) How can the results of the analytical model be validated using discrete event simulation?
- (3) How can such a system be optimally designed for a given set of parameters?

Based on the aim of this investigation and the research questions this paper intends to address, the following sub-targets can be formulated:

- An extensive literature review of analytical models in the context of multi-deep automated storage systems under different operational, particularly class-based, policies (Section 2).
- A thorough problem definition based on the research gap and the literature review (Section 3).
- A precise definition of the system under investigation, including a detailed process description (Section 4).
- The development of an analytical model for cycle time calculation (Section 5).
- The construction of a simulation model using a discrete event simulation (DES) for accurate validation (Section 6.1).
- A sensitivity analysis through comprehensive parameter variation (Section 6.2).
- An optimisation example to demonstrate the purpose and applicability of the analytical model (Section 6.3).

This analytical model represents the first straightforward and rapid tool for determining the cycle time and performance of a single robot. These results can be used as input for existing queueing models, such as those from Zou et al. (2018), Lehmann and de Koster (2024), or Trost and Eder (2024a). The consideration of demand-based sorted stacks is an important innovation both for academic research and practical applications, as the warehouse design process often assumes a random storage policy or relies on time-consuming and computationally intensive simulation models instead of analytical approaches that could also be used for optimisation.

This is a significant novelty since no existing approaches consider the impact of class-based policies in the context of RCS/RS. The main advantage is the provision of an easy

and fast tool to determine throughput. Hence, this analytical model could be particularly valuable for material handling providers, consultants, and customers. The limitation to a single robot can be justified by the complexity of the problem, and the fact that there are already analytical approaches, such as those by Zou et al. (2018) or Trost and Eder (2024a), which consider multiple robots and more than one picking station. These approaches could be utilised with input data from this study by applying this analytical model.

2. Literature review

The body of literature addressing RCS/RS is still quite limited. There are two notable analytical approaches discussing the performance of RCS/RS. The first is by Zou et al. (2018), who presented a semi-open queuing network (SOQN), and the second is by Trost and Eder (2024b), who developed a cycle time model (CTM) for estimating the performance of a single robot. Both studies validated their approaches by comparing the results with those from numerical simulations, which, alongside analytical formulas, is a common method for investigating such storage systems.

Zou et al. (2018) examined dedicated versus shared storage policies per stack and random versus zoned storage stacks along the grid. Their model can be used to determine the optimal width-to-length ratio and stack height. The key finding is that the dedicated policy enables higher throughput than the shared policy, although the latter is not favoured as it could result in up to twice the costs. Moreover, the significant advantage of the system's high space utilisation would not be fully leveraged. Despite the quality of the approximation, this analytical approach is neither easy nor quick to solve, which is one of the main research gaps, as companies continue to rely on numerical simulations to predict potential throughput. Chen (2022) and Tutam et al. (2024) also utilised a similar SOQN to discuss the performance of a RCS/RS. While Wang et al. (2023) investigated overhead robotic compact storage and retrieval systems (ORCS/RS), tested different storage policies, and compared ORCS/RS with the *AutoStore* system, Tutam et al. (2024) considered the impact of skewness in the design process of RCS/RS using Bender's curves.

Trost and Eder (2024b) developed an analytical calculation approach for RCS/RS with one robot serving multiple stack heights, assuming a uniform article distribution with uniform demand. These universally valid approximation formulas can accommodate various operational modes and a wide range of system parameters, such as robot velocity, grid size, stack height, container size, and the location of the I/O shaft.

In addition to these analytical approaches, several papers have conducted simulation studies on RCS/RS performance or control strategies. For instance, Tjeerdsma (2019), Galka and Scherbarth (2021), Kartnig et al. (2023), and Trost et al. (2023) developed discrete event simulations using a random storage strategy and a single class of items, while Beckschaefer et al. (2017) discussed different storage strategies. Chen et al. (2022) investigated various article classes to provide insights into the performance of ORCS/RS operating without return relocations. They derived the optimal system design for ORCS/RS and found that zoning results in better performance when demand is *Pareto* distributed.

Given the limited literature on RCS/RS, the research review was extended to include multiple-deep automated storage and retrieval systems (AS/RS) and multiple-deep shuttle-based storage and retrieval systems (SBS/RS) under class-based storage strategies. While RCS/RS systems excel in providing high storage densities, flexibility, and scalability, SBS/RS systems offer superior performance and faster order processing thanks to their use of numerous independent shuttles, each typically serving one tier within an aisle. As a result, SBS/RS systems are commonly employed in distribution and e-commerce warehouses. Conversely, AS/RS systems are primarily designed for larger storage units, such as pallets, and for handling heavier loads. However, since a single S/R machine serves an entire aisle, these systems have lower throughput rates and are generally used for applications with low turnover requirements. RMFS systems, such as the *Amazon Kiva* system, provide exceptional flexibility and scalability but come with the drawback of reduced storage densities. Table 2 summarizes the key characteristics of these warehouse types.

There are numerous papers discussing the performance of various storage systems under class-based storage policies, optimal storage layouts, and the best storage strategies. Therefore, the literature review in this context was limited to RCS/R systems and analytical approaches developed for performance approximation under class-based storage strategies.

The earliest approaches for AS/RS with class-based storage policies date back to the 1970s, such as Hausman et al. (1976), who investigated storage policies for AS/RS and compared the numerical results for random, two-class, and three-class storage strategies. Several further studies have been conducted on classical two-dimensional AS/RS. For instance, Petersen et al. (2004) compared the performance of class-based storage assignments with random storage assignments in a manual order-picking warehouse, finding that a class-based storage policy provides cost savings and enables higher performance. Yu and de Koster (2008) extended the research on class-based storage policies to 3D-AS/RS by developing an analytical approach. Zaerpour et al. (2013) further investigated 3D-AS/RS and found that the optimal storage dimension (excluding cuboid shapes) for a given number of storage slots is the same, regardless of which of the three storage policies mentioned is used. Yu et al. (2015) analysed the impact of using a finite number of items, concluding that more classes do not always result in better performance. Ekren et al. (2015) studied SBS/RS under class-based storage policies, considering different rack designs, and concluded that high-rise warehouse designs work well with class-based storage policies.

While Xu et al. (2018) examined multiple deep AS/RS with a two-class storage policy, Eder (2022) analysed storage strategies for multiple-deep SBS/RS with two and three classes, using a single queue with limited capacity (SQ-LC).

Table 2. Comparison of the different automated warehouse types.

Criterion	AS/RS	SBS/RS	RCS/RS	RMFS
Storage density	Medium	High	Very high	Medium
Turnover rate	Low-Medium	High	Very high	Medium
Item size	Large	Medium	Small	Medium
Investment cost	Low-Medium	High	Medium	Low-Medium
Flexibility	Low	High	Medium-High	Very high
Scalability	Low	High	High	Very high

Table 3. Literature overview.

Author	System	Policy	Model	DES ¹
Hausman et al. (1976)	AS/RS	CTM	random & class-based	
Petersen et al. (2004)	Warehouse ³	DES	random & class-based	
Yu and de Koster (2008)	3D-AS/RS	CTM	class-based	
Zaerpour et al. (2013)	3D-AS/RS	CTM	class-based	
Yu et al. (2015)	AS/RS	CTM	class-based	✓
Ekren et al. (2015)	SBS/RS	DES	class-based	
Beckschaefer et al. (2017)	RCS/RS	DES	random	
Xu et al. (2018)	AS/RS ²	CTM	random & class-based	✓
Zou et al. (2018)	RCS/RS	SOQN	random & class-based	✓
Galka and Scherbarth (2021)	RCS/RS	DES	random	
Eder (2022)	SBS/RS ²	SQ-LC	class-based	✓
Chen et al. (2022)	ORCS/RS	DES	random & class-based	
Trost et al. (2023)	RCS/RS	DES	random	
Kartnig et al. (2023)	RCS/RS	DES	random	
Wang et al. (2023)	ORCS/RS	SOQN	class-based	✓
Trost and Eder (2024b)	RCS/RS	CTM	random	✓
Lehmann and de Koster (2024)	AS/RS ²	CTM	random	✓
Trost and Eder (2024c)	RCS/RS	SQ-LC	random	✓
Trost and Eder (2024a)	RCS/RS	MQ-LC	random	✓
This paper	RCS/RS	CTM	random & class-based	✓

¹DES for validation, ² multiple-deep, ³ aisle-based manual picker warehouse.

Table 3 provides an overview of the existing scientific considerations.

As there are few studies investigating RCS/RS and no analytical approaches addressing the throughput of storage systems under class-based zoning across stack heights, this paper aims to present an analytical approach based on a cycle time model developed by Trost and Eder (2024b), taking into account the storage system's demand structure. This approach is novel, as Trost and Eder (2024b) assumed a uniform access structure across the stack height. A more detailed definition of the problem, along with the basic assumptions and the system under investigation, can be found in Sections 3 and 4.

3. Problem definition

In general, storage systems can be operated using one of the following storage assignment policies, as outlined by Hausman et al. (1976):

- Random (single class)
- Dedicated (one class for each product)
- Class-based (two or more classes)

A random storage policy, considering only a single class of items, assigns storage goods randomly to the first available location. In contrast, the dedicated storage policy reserves a specific storage location for each product (i.e. each class). The class-based storage policy classifies products into different classes based on criteria such as demand, cost, or size (Schenone et al., 2020).

A special case arises in the system depicted in this investigation. In the context of the policies discussed above, the storage policy within an RCS/R system involves random allocation along the grid. The first accessible stack with space for another container is

assigned, meaning that horizontal zoning along the grid is not applied. Therefore, high-demand A-articles are stored near the I/O shaft, while infrequently required C-articles are placed at the far edge of the grid. Although this arrangement may seem advantageous, horizontal zoning reduces performance because not all frequently accessed A-articles can be stored at the top of the grid. Previous studies, such as Trost and Eder (2024b), have demonstrated that the robot's ride time is minimal compared to the relocation times.

Compared to traditional automated storage systems (AS/RS), RCS/R systems can achieve high space utilisation rates (up to 400% increase) due to the absence of aisles and tiers. This is often accompanied by a potentially large number of relocations, depending on demand. Previous studies have shown that the number of relocations is significantly influenced by the stack height, in addition to the filling degree. RCS/R systems, such as the *Jungheinrich Powercube* or the *AutoStore* system, typically feature stack heights of up to 25 containers, which may result in a large number of relocations depending on demand distribution.

Furthermore, RCS/R systems are appealing for warehouse requirements such as scalability, flexible modularity, and easy expandability, though they do not necessarily guarantee high output performance. The potentially high number of necessary relocations can significantly impact throughput. This impact varies with article distributions featuring inhomogeneous access structures and the execution of return relocations, leading to demand-sorted stacks that can be classified into different access classes.

AutoStore asserts that 20% of the article stock in a warehouse generates 80% of the demand *Autostore* (2023). This empirical 80/20 rule is widely known as the *Pareto* principle.

From a logistical perspective, this means that 20% of the articles in a warehouse account for 80% of the demand. The impact of this phenomenon can be substantial, highlighting the importance of ensuring easy and fast access to high-demand articles. In this paper, the *Pareto* distribution is used as an example to discuss the performance of such systems. However, since demand may follow other probability distributions, alternative distributions can also be implemented.

Since every storage stock can be evaluated and analysed statistically, it is valuable to determine the performance of a storage system while considering the access structure. Based on an ABC analysis, a storage control system can effectively chart the demand for each product and identify the optimal access placement. This is achieved by arranging data according to predefined order criteria in descending order from left to right Brüggenmann and Bremer (2020).

Material handling providers still simulate nearly every new storage system before it is sold to the customer. Although this investigation presents an analytical model and highlights the advantages of this method – such as simplicity, clarity, and computational efficiency – it is important to also acknowledge its drawbacks. Analytical models often rely on simplifying assumptions to generate solvable equations, making them less effective for describing highly complex systems. Furthermore, they are not well-suited for systems that evolve over time, especially when dynamic simulation of system behaviour is required.

As mentioned in the introduction, the disadvantages often associated with simulation models include high computational demands and a lack of generalisability. In contrast, simulations can effectively model real-world scenarios and are

particularly useful for visualisation, which is employed for simulation verification in this study. Frequently, both methods are used together in a complementary manner: an analytical model can provide valuable insights or serve as a benchmark, while simulations are better suited for handling more complex or realistic scenarios.

An analytical approach that considers a demand-based access structure represents a novel development. Such a model could quickly yield appropriate results within a few seconds for a given article distribution and parameter setting.

When designing a new storage system, the three main input parameters are the required storage capacity, the desired throughput, and the existing article demand structure.

The access structure must be analysed and categorised into different classes, such as one class for each stack height or three classes following the ABC principle. Based on this categorisation, a discrete probability distribution that aligns well with the demand structure must be identified. Maximum likelihood estimation could be one method for this purpose.

This investigation focuses on throughput depending on the applied distribution and its impact on the probability of relocation. The approach developed in this paper can also be easily modified and applied to other storage systems.

Building on Trost and Eder (2024b), this paper adapts and extends the developed cycle time model (CTM) to consider an inhomogeneous access structure with any demand distribution. The *Pareto* distribution is used as an example, but any discrete demand distribution can also be applied. This approach can be used to determine the throughput of an RCS/R system operating with one robot and return relocations (e.g. *AutoStore*). Return relocations, while elongating the cycle time, also ensure a self sorting effect over the stack. The access structure must be analysed and categorised into different classes, such as one class for each stack height. Based on this, a suitable discrete probability distribution must be identified.

The primary advantage of this approach is its simplicity and speed in determining the cycle time or throughput of an RCS/RS using a class-based access structure. It can be used with a given article demand that has already been analysed and assigned to a specific distribution, or with theoretical relocation probabilities, such as average digging depth.

4. System description

The system depicted in this paper is a block-arranged, grid-based storage system operated from above by a robot. This configuration enables fully autonomous storage and retrieval. Inside the storage system, goods are stored in plastic containers stacked up to a height of 25 containers. Consequently, the storage strategy within each stack follows the Last-In-First-Out (LIFO) principle. The main advantages of the system are its high storage density and scalability. However, due to the high stack heights, the number of necessary relocations can also be significant. Figure 2 illustrates an RCS/RS with one robot serving a single picking station.

Figure 2 illustrates an RCS/RS. A picking station with an input/output (I/O) shaft is positioned at the center of the grid's wide edge. The robot operates on the grid to manage the storage system. Such storage systems typically use a random storage

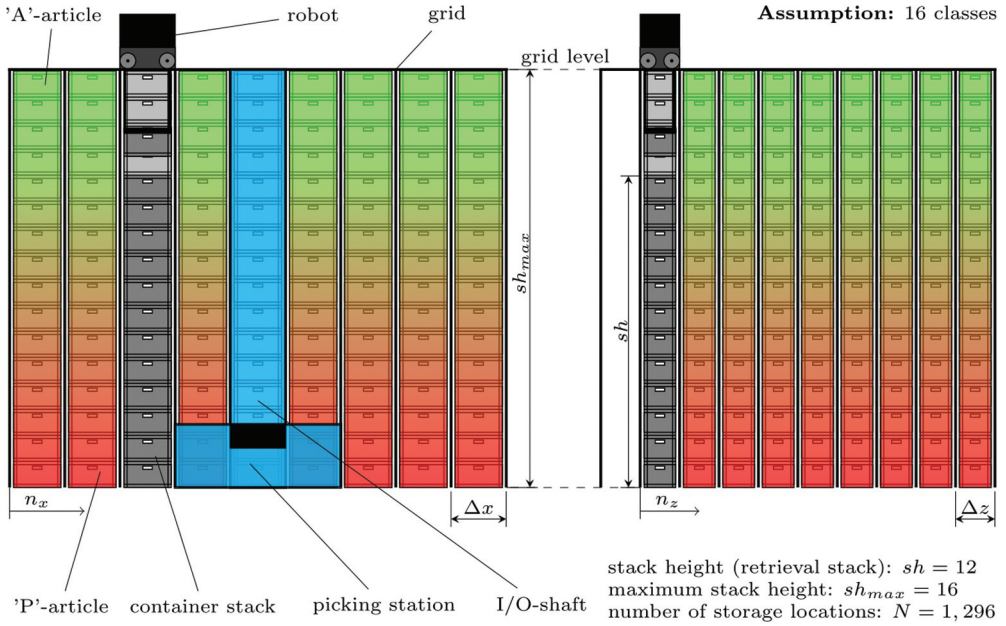


Figure 2. Robotic compact storage and retrieval system.

strategy combined with a ‘relocation to next available stack’ relocation policy. A class-based storage strategy can optimise throughput by reducing the robot’s ride time on the grid.

Additionally, some systems employ return relocations to achieve self sorted stacks, effectively implementing a class-based storage strategy within each stack. Return relocations involve returning every relocated container back to its original stack in the reverse sequence immediately after the required container is retrieved. The retrieved container is then placed on top of any accessible stack within the grid. If this container is needed soon afterwards, the necessary relocations to access it are minimal or even zero. Conversely, if the container is not needed for an extended period, it will be moved down as other containers are placed above it. This approach naturally results in demand-sorted stacks: containers with high-demand articles are frequently retrieved and consistently placed at the top, while less frequently accessed containers gradually shift to the bottom over time. A more detailed description of the storage and retrieval processes, with or without return relocations, including graphical illustrations, can be found in Trost and Eder (2024b).

Unlike vertical zoning facilitated by return relocations, horizontal zoning along the grid is uncommon. This is because the robot’s ride time is relatively short compared to the lengthy container lifting and lowering times and the potentially high relocation times due to the numerous relocations required. Therefore, reducing relocations through vertical zoning can enhance performance (Kartnig et al., 2023). Although an additional horizontal zoning approach could be incorporated into the analytical model, it is not implemented here due to its limited practical relevance.

Figure 3 illustrates the storage and retrieval cycle, including a relocation and a return relocation, performed in a DCC.

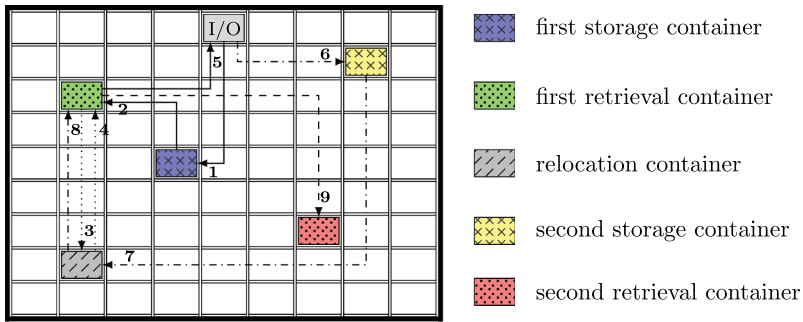


Figure 3. Overview of the operation cycle performing DCC with return relocations.

The numbers describe the sequence of the robot's movements. Starting with a container to be stored (blue, path 1), this is followed by a retrieval (green, path 2). Consequently, the relocation container must be moved to a free relocation stack (paths 3 and 4). The retrieval container can then be transported to the I/O shaft via path 5. Once the retrieval process is complete, the robot picks up another storage container (yellow, path 6). After storing this container, the robot rides to the previous relocation stack (path 7) and returns the previously relocated container (path 8). The next retrieval order can then be processed.

5. Analytical approach

This paper's analytical approach builds on the work of Trost and Eder (2024b), who introduced the first straightforward calculation tool to predict the throughput of a single robot within an RCS/RS for any system configuration, irrespective of parameters such as warehouse size, stack height, filling degree, container type, or kinematic data. This approach served as the foundation for further developments that extended to multiple robots and picking stations. The method of re-utilising and refining the existing CTM and queueing models from related storage systems, such as AS/RS or SBS/RS, has ensured high consistency in the results.

Trost and Eder (2024b), assumed a uniform access structure over the stack height. However, this may not reflect real-world applications, such as *AutoStore*, where many RCS/R systems feature an inhomogeneous access structure and leverage the beneficial self sorting effect on throughput through return relocations. This paper extends the equations from Trost and Eder (2024b), to calculate the performance of storage systems with an inhomogeneous demand distribution. The main assumptions are listed below:

- The robot works in a dual command cycle under the First-Come-First-Served rule.
- The systems I/O point is in front of the I/O shaft.
- The I/O shaft is located in the middle of one of the grid's edges.
- There are always totes waiting at the I/O point in front of the I/O shaft. This assumption is necessary for maximum throughput calculations.
- The robot picks up a new container to be stored after dropping off a required container.

- The robot's velocity is constant. If not, a realistic velocity rate has to be calculated.
- The container to be relocated is relocated to the nearest available storage location using an 'one-path' relocation policy.
- The filling degree is limited to a specific value to ensure that relocations can be done.
- The maximum number of classes equals the maximum stack height.
- The order of a container from different classes is given by the storage policy.
- The order of a container from the same class is evenly distributed over all containers of the same class.
- Once a container is assigned to a class, the assignment is fixed and can only be changed after retrieval.

The cycle time calculation in this paper is divided into the following steps: calculating the robot's ride time (section 5.1) for storage, relocation, and retrieval; determining the probability of a relocation (section 5.2); assessing the lifting and lowering times (section 5.3); and evaluating the service time at the I/O shaft (section 5.4). Table 4 presents the notations used in this paper.

The cycle time of a robot can vary depending on the operation mode. This paper assumes an inhomogeneously distributed demand and return relocations to achieve demand-based sorted stacks. Therefore, the expected cycle time for a combined storage and retrieval process, including return relocations in a DCC, is calculated according to Trost and Eder (2024b), by Equation 1:

Table 4. Notation.

a	Distribution parameter
Δx	Distance between two grid elements along the x-axis
Δz	Distance between two grid elements along the z-axis
a_R	Robot's acceleration rate for horizontal ride
a_T	Robot's acceleration rate for container transfer
$E(CT)$	Expectation of the cycle time
f	Filling degree
G	Gini coefficient
h_C	Height of a storage container
k_0	Position of the picking station along the x-axis
n_C	Number of classes
n_x	Number of stacks along the x-axis
n_z	Number of stack along the z-axis
n_{St}	Number of stacks
p_{rel}	Probability of a relocation
sh	Storage height of a container stack
t_{CX}	Time for the container exchange at the picking station
t_L	Time to open/close the locking claws
t_{IO_DCC}	Time required at the I/O shaft in a dual command cycle
t_{R_DCC}	Additional time of a robot to ride in a DCC
t_{R_SCC}	Time of a robot required to ride in an SCC
t_{R_rel}	Time of a robot required to ride at the relocation cycle
t_T	Time required to transfer a container
t_{T_rel}	Time required to transfer a container in the relocation cycle
t_{WX}	Time of a robot to change the wheels from one direction to another
v_R	Velocity rate of a robot in horizontal direction
v_T	Velocity rate of a robot for lifting and lowering
w_{rel}	Number of necessary relocations per retrieval
w_{rel_uni}	Number of relocations per retrieval assuming a uniform distribution
y	Stack height variable

$$E(CT) = 2 \cdot (t_{R_SCC} + t_{R_DCC} + t_T + w_{rel} \cdot (t_{R_rel} + 2 \cdot t_{T_rel})) + t_{IO_DCC} \quad (1)$$

In this context, t_{R_SCC} represents the robot's ride time in an SCC, while t_{R_DCC} denotes the time required for operation in a DCC, specifically the ride from one stack to another. Both of these times occur twice in a DCC with return relocations. The term t_T calculates the lifting and lowering time for storage and retrieval. Additionally, the expression $w_{rel} \cdot (t_{R_rel} + 2 \cdot t_{T_rel})$ represents the probability of a relocation multiplied by the time required for the relocation cycle.

To account for realistic riding characteristics, both acceleration and deceleration must be considered for horizontal rides and vertical container transport. Equation 2 distinguishes between trapezoidal and triangular motion profiles. Depending on the distance to be covered, the robot may reach maximum velocity and ride at a constant speed until deceleration begins, or it may not reach maximum velocity. For $j < \frac{v_R^2}{a_R}$, the maximum velocity will not be reached, and the first equation in Equation 2 should be used Trost and Eder (2024b).

$$t(j) = \begin{cases} 2 \cdot \sqrt{\frac{j}{a_R}} & \text{for } j < \frac{v_R^2}{a_R} \\ \frac{j}{v_R} + \frac{v_R}{a_R} & \text{for } j \geq \frac{v_R^2}{a_R} \end{cases} \quad (2)$$

Analogous to the horizontal ride, the vertical lifting and lowering of the containers also take into account the acceleration of the robot's transfer device. Equation 3 presents the case distinction.

$$g(y) = \begin{cases} 2 \cdot \sqrt{\frac{y}{a_T}} & \text{for } y < \frac{v_T^2}{a_T} \\ \frac{y}{v_T} + \frac{v_T}{a_T} & \text{for } y \geq \frac{v_T^2}{a_T} \end{cases} \quad (3)$$

If the height y is smaller than $\frac{v_T^2}{a_T}$, the lifting device accelerates only up to half the height before beginning to decelerate, operating in triangular mode. For heights greater than or equal to $\frac{v_T^2}{a_T}$, the device reaches the maximum velocity v_T and operates in trapezoidal mode.

5.1. Robot ride time

The mean ride time of a robot operating in an SCC can be calculated by Equation 4 Trost and Eder (2024b). The *sign* function accounts for possible wheel exchanges by evaluating all possible combinations of storage locations along the grid in the x- and z-directions.

$$t_{R_SCC} = \frac{1}{n_x} \cdot \frac{1}{n_z} \cdot \sum_{k=1}^{n_x} \sum_{l=1}^{n_z} t(l \cdot \Delta z) + t((|k - k_0|) \cdot \Delta x) + t_{WX} \cdot \mathbf{sign}(|k - k_0|) \quad (4)$$

Analogous to the ride time in an SCC, the robot's ride in a DCC includes an additional term (Equation 5) that represents the robot's ride from a storage stack to a retrieval stack Trost and Eder (2024b).

$$t_{R_DCC} = \frac{1}{n_x^z} \cdot \frac{1}{n_z^z} \cdot \sum_{k=1}^{n_x} \sum_{l=1}^{n_x} \sum_{m=1}^{n_z} \sum_{n=1}^{n_z} t((|m-n|) \cdot \Delta z) + t((|k-l|) \cdot \Delta x) + t_{WX} \cdot \mathbf{sign}((|k-l|) \cdot (|m-n|)) \quad (5)$$

For cases where $sh > 1$, indicating that relocations are necessary, Trost and Eder (2024c) provide a closed-form expression for the ride time during a relocation cycle. Their model assumes a circular relocation strategy, where the container to be relocated is transported to the nearest stack with available space. This paper adopts the relocation strategy described by Trost and Eder (2024c), and the robot's ride time $t_{R,el}$ for a relocation cycle is calculated using the same equations as those presented in Trost and Eder (2024c),

5.2. Relocation probability

For $sh > 1$, and with an arbitrarily distributed article access structure, a specific relocation probability can be derived. In this investigation, we assume that the demand follows a *Pareto* distribution over the stack height sh . The density function for a variable y , with parameters $y_{min} > 0$ and $\alpha > 0$, is given by Schlittgen (2008) and depicted by Equation 6:

$$f(y) = \frac{\alpha \cdot y_{min}^\alpha}{y^{\alpha+1}} \quad (6)$$

And the distribution function is presented by Equation 7 (Schlittgen, 2008):

$$F(y) = 1 - \left(\frac{y_{min}}{y} \right)^\alpha \quad (7)$$

Based on Equation 6, the probability of a relocation for each position n along the stack height sh can be calculated by the following expression (Equation 8):

$$p(y) = \frac{y^{-(\alpha+1)}}{\sum_{n=1}^{sh} y^{-(\alpha+1)}} \quad (8)$$

If a different probability distribution is assumed, Equations 6 to 8 must be adapted accordingly. The number of necessary relocations required to retrieve one container, given a discrete probability distribution, can be determined as follows (Equation 9):

$$w_{rel} = \sum_{i=0}^{sh-1} p(y) \cdot (i-1) \quad (9)$$

Based on Trost and Eder (2024b), the number of relocations per retrieval, assuming a uniform access structure over the stack height, can be calculated using the formula below (Equation 10). This calculation depends on the filling degree f and the stack height sh :

$$w_{rel_uni} = \sum_{n=0}^{sh-2} \sum_{i=1}^{sh-1-n} \cdot \frac{i}{sh-n} \cdot \binom{sh}{n} \cdot f^{sh-n} \cdot (1-f)^n \quad (10)$$

Finally, building the ratio ξ_{rel} (Equation 11):

$$\xi_{rel} = \begin{cases} 1 & \text{for } sh = 1 \\ \frac{1 \cdot w_{rel}}{f \cdot w_{rel_uni}} & \text{for } sh \geq 2 \end{cases} \quad (11)$$

This factor adjusts the influence of digging depth in the case of relocations. Under a uniform distribution assumption, a storage container would typically be found about halfway up the stack, especially in a nearly full warehouse. However, with an inhomogeneous distribution, such as the *Pareto* distribution, containers are more likely to be located higher in the stack. The factor ξ accounts for this adjustment by reflecting the change in the expected container location due to the skewed distribution.

5.3. Container transfer time

The mean time required for lifting and lowering containers during storage or retrieval can be calculated by Equation 12 using the following slightly modified expression from Trost and Eder (2024b), incorporating the previously defined factor ξ :

$$t_T = t_L + \xi_{rel} \cdot \sum_{n=1}^{sh} \sum_{i=0}^{n-1} \frac{1}{sh+4 \cdot i} \cdot \binom{sh-1}{i} \cdot f^{sh-1-i} \cdot (1-f)^i \cdot 2 \cdot g(h_C \cdot n) \quad (12)$$

While the first term, t_L , represents the fixed time required for locking and unlocking the container, while the second term describes the time needed for lifting and lowering, which depends on the stack height (sh), the filling degree (f), the container height (h_C), the velocity (v_T). Additionally, it accounts for the influence of the access structure and digging depth. The g function provides the correct transport time.

The mean time for lifting and lowering during a relocation cycle is calculated using Equation 13:

$$t_{T_rel} = t_L + \sum_{n=1}^{sh} \sum_{i=0}^{n-1} \frac{1}{sh+4 \cdot i} \cdot \binom{sh-1}{i} \cdot f^{sh-1-i} \cdot (1-f)^i \cdot 2 \cdot g(h_C \cdot n) \quad (13)$$

5.4. Picking time

The picking time consists of the time the robot waits on the I/O shaft, the lifting and lowering time through the I/O shaft to the picking station, and the container exchange time via a conveyor or a rotating plate.

The following Equation 14 is derived from Trost and Eder (2024b) and has been adapted for the picking process in a DCC, taking into account the acceleration and deceleration of the lifting device.

$$t_{IO_DCC} = 2 \cdot t_L + t_{CX} + 2 \cdot g(h_C \cdot sh) \quad (14)$$

Therein, t_L represents the time for opening and closing the locking claws, t_{CX} denotes the container exchange time within the picking station, and the g function provides the correct vertical container transport time based on the container height and the stack height.

6. Numerical study

In addition to storage capacity, throughput is a key parameter when designing a new storage system. For an inhomogeneous demand distribution, such as the classical *Pareto* principle where 20% of the storage items account for 80% of the requisitions, the access structure significantly impacts throughput. Therefore, the article demand structure is also a crucial factor in warehouse design.

To analyse throughput in relation to the access structure and to validate the analytical approach outlined in [section 5](#), a discrete event simulation (DES) was developed. This simulation re-models an RCS/R system and its processes using the DES software SIMIO (version 15.240).

[Table 5](#) presents the parameters used in the numerical study. All values are derived from European material handling providers and represent standard values for RCS/R systems.

The primary assumption is that containers are distributed according to the *Pareto* principle across the stack height sh . This paper initially uses a single class for each stack height with the corresponding access distribution, but it is also possible to consider multiple classes over the stack height, such as AB or ABC.

To validate the analytical approach outlined in [section 5](#), results will be compared with data from 30 independent scenarios of the simulation model, each involving 11,000 order retrievals. The first 1,000 retrievals are excluded to eliminate the transient phase. The simulation experiments range from a few minutes to several hours in duration.

The simulation was verified to ensure that the model accurately represents the conceptual design and has been correctly implemented. To achieve this, the simulation logic and behaviour within the simulation environment were peer-reviewed to confirm that all assumptions were properly applied in the code. Additionally, each simulation process was tested independently, and a sensitivity analysis was conducted to evaluate whether the outputs responded as expected to variations in input parameters.

The simulation model is built around the four core processes of an RCS/R system: storage, retrieval, relocations, and return relocations. In the simulation, the robot is assigned to an order and rides to the designated stack. If containers are stored above

Table 5. Parameters for the RCS/RS.

Parameter	Value
Access structure	<i>Pareto</i> distributed
Horizontal distance in x	$\Delta x = 0.7m$
Horizontal distance in z	$\Delta z = 0.5m$
Container exchange time	$t_{CX} = 5s$
Container height	$h_C = 330mm$
Filling degree	$f \in \{85\%, 90\%, 95\%, 96\%, 97\%, 98\%\}$
Number of grid elements along the x-axis	$n_x \in \{10, 15, \dots, 50\}$
Number of grid elements along the z-axis	$n_z \in \{10, 15, \dots, 50\}$
Position of the I/O shaft along the x-axis	$k_0 = n_x/2$
Robot horizontal acceleration rate	$a_R = 0.8ms^{-2}$
Robot vertical acceleration rate	$a_T = 2.5ms^{-2}$
Robot horizontal velocity rate	$v_R = 3.1ms$
Robot lifting and lowering velocity rate	$v_T = 1.6ms$
Robot time to lock/unlock the container	$t_L = 1s$
Robot wheel change time	$t_{WX} = 1s$
Storage height of a container stack	$sh \in \{1, 2, \dots, 25\}$

the required one, the robot relocates them to free neighbouring stacks. After accessing the needed container, the robot transports it to the I/O shaft. Once the container is lowered through the I/O shaft, the robot picks up another container from the picking station and stores it on any available stack. Finally, the robot return relocates all previously moved containers in reverse order to the original stack from which the retrieval was made. The order process is complete when the stack's sorted sequence (excluding the retrieved container) is restored.

6.1. Validation of the analytical approach

Validating the analytical approach is a crucial step to ensure that the analytical results align with the outputs from the simulation model under comparable conditions. This is done by comparing the simulation results with those obtained from the analytical model.

For this study, a factorial approach will be employed to systematically validate the analytical model and examine the effects of varying several parameters: filling degree f , number of container stacks $n_{St} = n_x \cdot n_z$, stack height sh , and the distribution parameter α . Each parameter will be tested at multiple levels to capture both main effects and interactions.

The filling degree will be varied across different percentage levels to assess its impact on system efficiency. The number of stacks will be adjusted to evaluate its effect on classification accuracy and throughput. Stack height will be manipulated to understand how vertical dimensioning influences stability and performance. Finally, the distribution parameter α will be changed to examine its role in the distribution of demands across the stack height.

Starting with the assumption of a *Pareto*-distributed demand over a stack height of $sh = 10$, corresponding to 10 classes, Figure 4 shows a *Pareto* chart with the relative and cumulative frequencies plotted over the stack height using the parameters $\alpha = 0.1$ and $y_{min} = 1$.

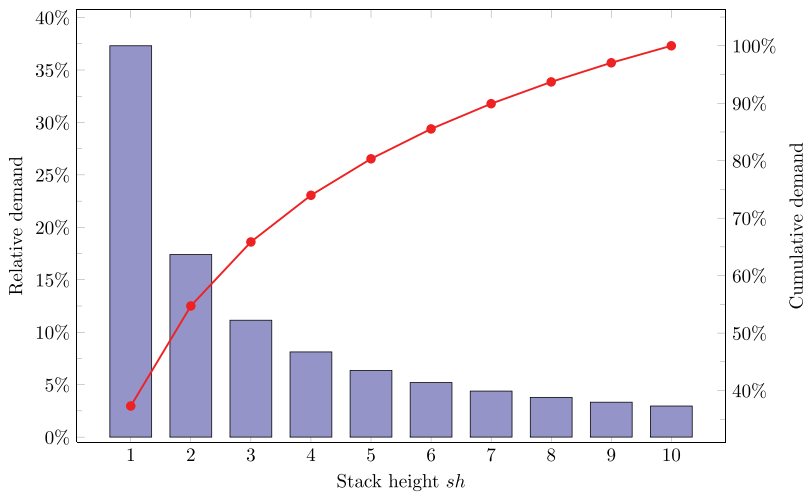


Figure 4. Example of a *pareto* plot with the parameters $\alpha = 0.1$ and $y_{min} = 1$.

Interpreting the plot from Figure 4, approximately 54% of the demand can be met with articles stored in the first or second stack height from the top. Furthermore, the inequality is even more pronounced in this case: over 80% of the requests can be fulfilled by accessing only the top 50% of the stack height. Accessing containers on the ground floor is necessary for only about 3% of the total demand. In the literature, this scenario is described as 54/20, indicating that 54% of the demand can be met by accessing just 20% of the total stack height.

These factors impact both the relocation probability and the overall system performance. To understand the extent of this impact and to validate the accuracy of the presented analytical model, this section focuses on throughput and relocation probability.

Figure 5 illustrates the throughput ϑ as a function of the filling degree f . It compares results from the analytical approach (red curve) with those from the DES (blue curve). Additionally, results assuming a uniform distribution are shown (brown curve), highlighting the effect of a non-uniform distribution. The filling degree is varied within a practically relevant range from 85% to 98%.

As shown in Figure 5, the impact of the filling degree on throughput is relatively small. However, once the filling degree reaches $f = 95\%$, the throughput curve starts to decline more sharply. A fully stocked warehouse results in longer relocation ride times and, consequently, extended relocation cycles. Compared to a uniform distribution, the *Pareto* distribution provides better performance.

Table 6 presents all the data, including the minor estimation errors.

Figure 6 displays the throughput ϑ as a function of the number of stacks n_{St} .

Figure 6 illustrates how grid size influences throughput, comparing a *Pareto*-distributed demand to a uniform access structure. The results show that the *Pareto*-distributed access structure (54/20) delivers approximately 20% higher performance. The accuracy of the analytical approach is validated as reliable, as evidenced by the error rates detailed in Table 7.

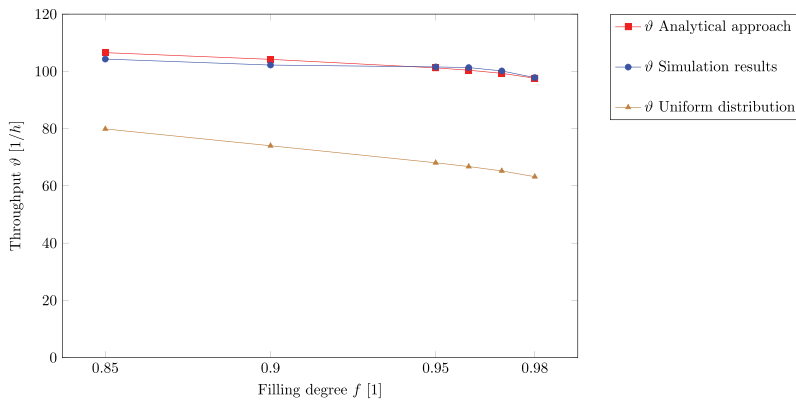
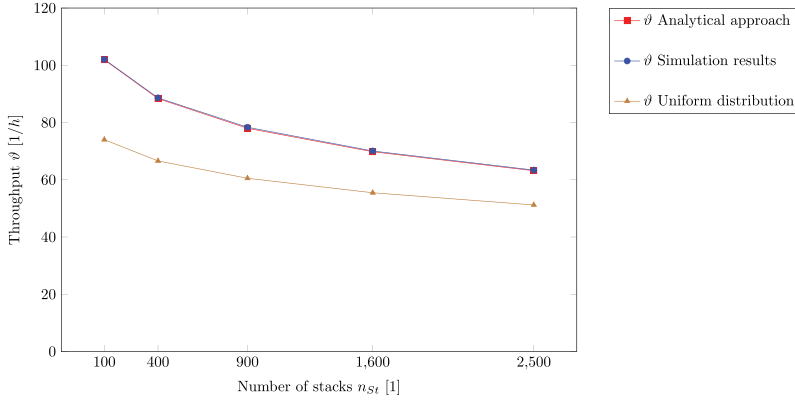


Figure 5. Throughput ϑ depending the filling degree f of a 10 by 10 RCS/RS with a stack height of $sh = 10$ (distribution parameter $\alpha = 0.1$).

Table 6. Throughput depending on the filling degree f and the stack height n_{st} for a stack height of $sh = 10$.

		Analytical	Simulation	Error	Uniform
		ϑ [1h]	ϑ [1h]	[%]	ϑ [1h]
f	85%	106.542	104.296	2.153%	79.892
	90%	104.196	102.205	1.948%	74.013
	95%	101.224	101.554	-0.325%	68.069
	96%	100.406	101.319	-0.902%	66.746
	97%	99.315	100.159	-0.842%	65.217
	98%	97.615	97.807	-0.196%	63.207

**Figure 6.** Throughput ϑ depending the number of stacks n_{st} of an RCS/RS with a stack height of $sh = 10$ and a filling degree of $f = 90\%$ (distribution parameter $\alpha = 0.1$).**Table 7.** Throughput depending on the filling degree f and the stack height n_{st} for a stack height of $sh = 10$.

		Analytical	Simulation	Error	Uniform
		ϑ [1h]	ϑ [1h]	[%]	ϑ [1h]
n_{st}	100	104.196	102.205	1.948%	74.013
	400	88.621	88.362	0.293%	66.587
	900	78.351	77.989	0.464%	60.520
	1600	70.055	69.796	0.371%	55.468
	2500	63.387	63.162	0.356%	51.195

Based on the previous analysis, the effect of stack height and distribution choice on system performance will be examined. For this investigation, the parameter is set to $\alpha = 0.6$, which results in an 80/20 distribution with a stack height of $sh = 20$.

Figure 7 compares the results from the analytical approach with those obtained from discrete event simulation. It shows how throughput ϑ and relocation probability w_{rel} vary with stack height sh .

The throughput ϑ decreases for an increasing stack height sh . Hence, the throughput ϑ decreases with increasing inequality. This can be explained by the increasing stack height sh while the parameter of the distribution α keeps constantly. Figure 7 shows a good approximation quality both for the throughput as well as for the relocation probability. The biggest estimation error is smaller than 4%. All the results and the estimation errors as well as the Gini coefficient are listed in Table 11.

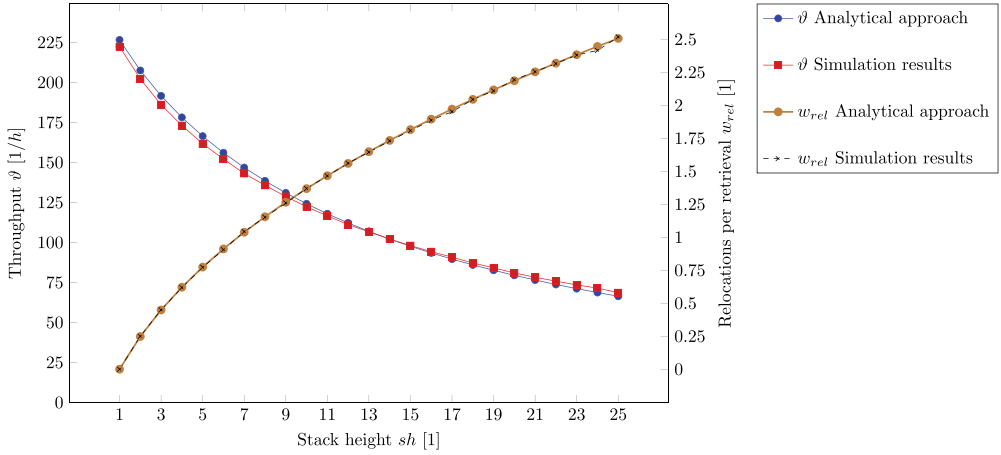


Figure 7. Throughput and relocation probability depending on the stack height sh for a grid size of 10 by 10 and a parameter $\alpha = 0.6$.

To gain further insights and continue evaluating the analytical approach, the parameter α , previously considered fixed, will now be varied within a range that is realistic for RCS/R systems. Figure 8 shows the throughput and relocation probability for two different stack heights as functions of the parameter α . Additionally, the results from the analytical model are compared with those obtained from the DES.

Figure 8 illustrates that throughput increases almost linearly with the parameter α , though the curve flattens significantly for values above $\alpha = 1.5$, regardless of stack height sh . The relocation probability w_{rel} shows an inverse relationship with α , decreasing monotonically. As α and stack height sh increase, the distribution becomes more unequal, leading to greater disparity.

Table 10 provides all the data presented in Figure 8, including estimation errors.

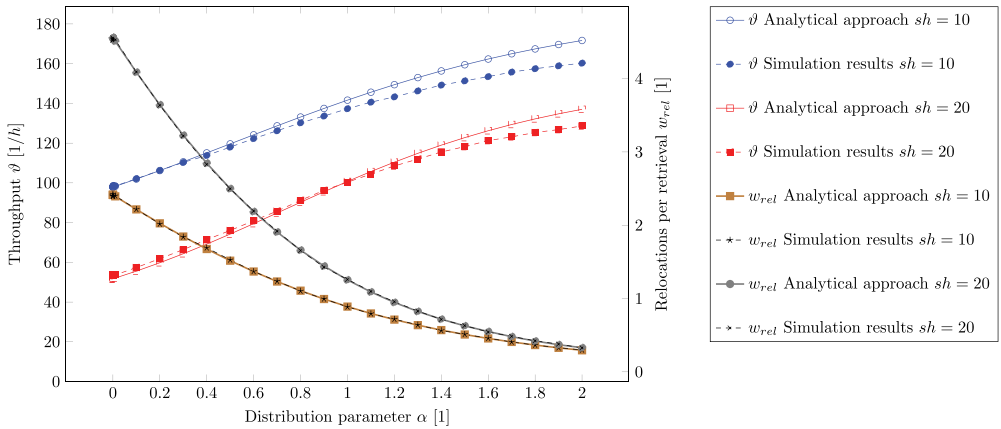


Figure 8. Throughput and relocation probability depending on the parameter α for a grid size of 10 by 10 and the two stack heights $sh = 10$ and $sh = 20$.

The most significant estimation error occurs at high α values for throughput, but remains under 8%. This is due to the low number of relocations required; for example, with $\alpha = 0.1$ and a stack height of $sh = 20$, an average of more than four relocations per retrieval is needed, whereas for $\alpha = 2$, the relocation probability is about 0.4. Since each simulation run involves 10,000 retrievals, the number of relocations is quite low. To achieve more accurate results for high α values, approximately 100,000 retrievals would be necessary. Overall, the calculated relocation probability aligns well with the results from the DES.

6.2. Parameter variation

This subsection aims to demonstrate the breadth and scope of the presented approach. Figure 9 shows the relocation probability plotted against stack height sh and the parameter α .

The evaluated data can be found in Table 12.

As illustrated, the *Pareto* distribution encompasses a wide range of relocation probabilities, starting from higher probabilities at minimal parameters α . As α increases, the relocation probability w_{rel} decreases, regardless of the stack height sh .

Similarly, it is important to determine how throughput ϑ varies with the parameter α and stack height sh . For this analysis, the grid size of the RCS/RS is set to 10 by 10, equating to 100 stacks, and the maximum filling degree is $f = 90\%$.

Figure 10 shows the throughput ϑ for a 10 by 10 RCS/R system with one robot serving one picking station, plotted against stack height sh and the parameter α .

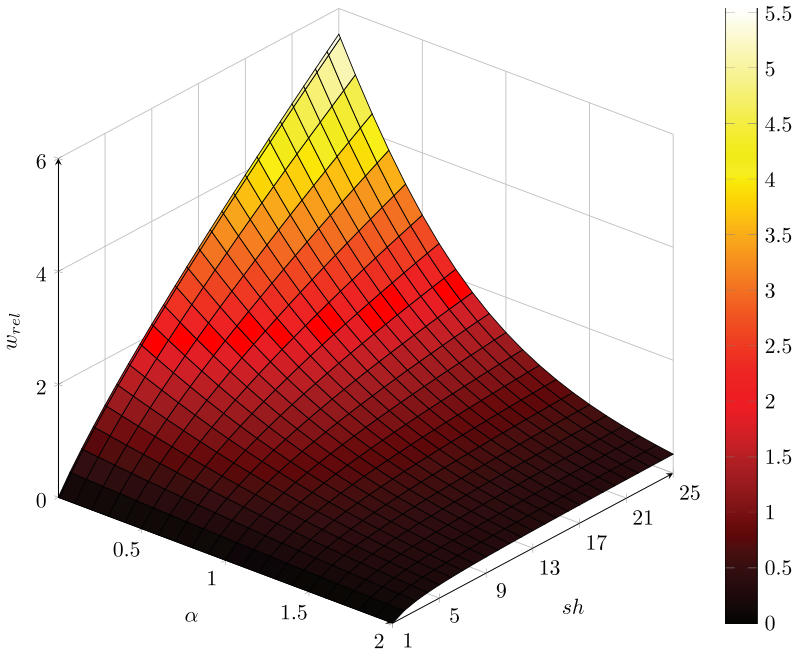


Figure 9. Relocation probability w_{rel} plotted over the stack height sh and the parameter α .

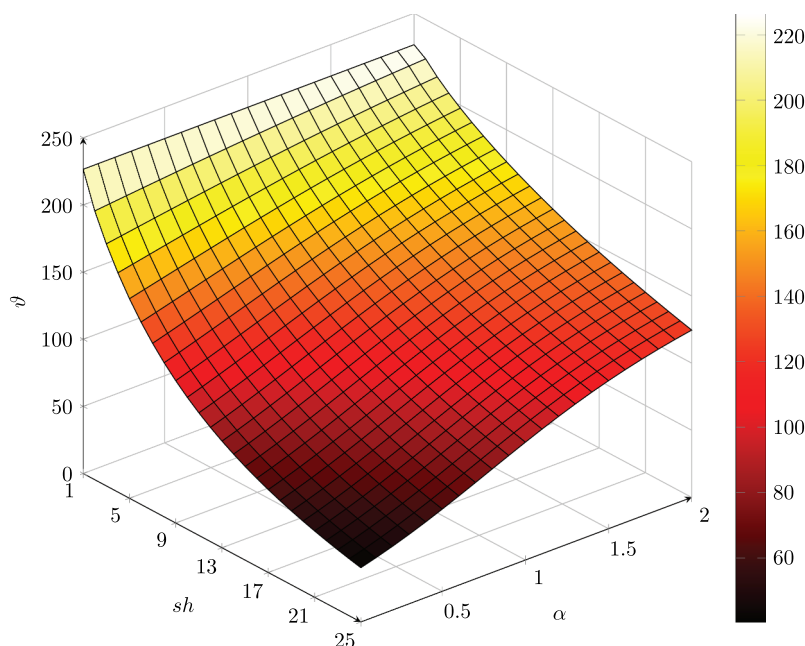


Figure 10. Throughput of an RCS/RS depending on the distribution parameter α and the stack height sh for a grid size of 10 by 10.

As observed, the throughput ϑ increases almost linearly with rising parameter α , while it exhibits distinct convex parabolic characteristics as stack height increases. High stack heights combined with highly scattered demands, indicated by low values of α , result in lower throughputs. Conversely, a high level of inequality – reflected by a high parameter α - leads to higher performance due to fewer required relocations.

6.3. Application example

This section aims to demonstrate the relevance of this approach for warehouse design. It involves defining specific requirements for a storage system and identifying the optimal system based on various parameters. Table 8 provides the input data needed for the application example.

The storage system is required to have a capacity of 10,000 containers. The demand structure, which should be analysed based on the evaluated customer data, can be

Table 8. Input data for the application example.

Parameter	Value
Required storage capacity	$N = 10,000 (\pm 2\%)$
Number of stacks along x-axis	$n_x \in \{10, 15, \dots, 50\}$
Number of stacks along z-axis	$n_z \in \{10, 15, \dots, 50\}$
Storage height of a container stack	$sh \in \{6, 7, \dots, 25\}$
Filling degree of the storage system	$f = 90\%$
Throughput required	$\vartheta_{requ} = 80 \text{ 1h}$
Demand distribution	80/20

Table 9. Application example for RCS/RS with one robot.

no	$n_x \times n_z$	sh	N	α	ϑ [1h]	A [m ²]	V [m ³]
1	40x10	25	10000	0.518	56.128	196.0	2,009
2	20x20	25	10000	0.518	56.865	168.0	1,722
3	50x10	20	10000	0.600	67.076	245.0	2,107
4	25x20	20	10000	0.600	69.494	210.0	1,806
5	45x15	15	10125	0.728	82.771	299.3	2,080
6	50x20	10	10000	0.969	96.460	420.0	2,226
7	40x25	10	10000	0.969	97.548	406.0	2,152

Table 10. Comparison of the results throughput ϑ and relocation probability w_{rel} analytical approach versus discrete event simulation for a grid size of 10 by 10 and a stack height of $sh = 10$ and $sh = 20$

		Analytical approach				Discrete event simulation				Estimation error			
		$sh = 20$		$sh = 10$		$sh = 20$		$sh = 10$		$sh = 20$		$sh = 10$	
		ϑ [1h]	w_{rel}	ϑ [1h]	w_{rel}	ϑ [1h]	w_{rel}	ϑ [1h]	w_{rel}	ϑ [1h]	w_{rel}	ϑ [1h]	w_{rel}
α	10^{-6}	51.724	4.559	97.838	2.414	53.720	4.527	98.492	2.404	3.72%	-0.72%	0.66%	-0.41%
	10^{-5}	51.724	4.559	97.838	2.414	53.763	4.523	98.652	2.397	3.79%	-0.80%	0.83%	-0.71%
	10^{-4}	51.728	4.559	97.841	2.414	53.491	4.554	98.100	2.424	3.30%	-0.10%	0.26%	0.43%
	10^{-3}	51.761	4.554	97.878	2.412	53.615	4.539	98.597	2.397	3.46%	-0.33%	0.73%	-0.63%
	10^{-2}	52.094	4.511	98.239	2.394	53.655	4.530	98.425	2.401	2.91%	0.42%	0.19%	0.29%
	10^{-1}	55.581	4.088	101.936	2.216	57.385	4.091	102.071	2.218	3.14%	0.09%	0.13%	0.08%
	0.2	59.788	3.643	106.199	2.026	61.859	3.637	106.240	2.017	3.35%	-0.15%	0.04%	-0.44%
	0.3	64.325	3.228	110.594	1.846	66.396	3.219	110.299	1.840	3.12%	-0.28%	-0.27%	-0.30%
	0.4	69.156	2.846	115.083	1.675	71.414	2.831	113.793	1.693	3.16%	-0.55%	-1.13%	1.02%
	0.5	74.229	2.499	119.625	1.516	76.053	2.498	117.994	1.528	2.40%	-0.05%	-1.38%	0.73%
	0.6	79.480	2.186	124.177	1.369	80.949	2.194	122.277	1.370	1.81%	0.35%	-1.55%	0.09%
	0.7	84.834	1.907	128.694	1.232	85.791	1.919	126.215	1.237	1.12%	0.63%	-1.96%	0.38%
	0.8	90.210	1.660	133.134	1.107	91.216	1.655	130.218	1.102	1.10%	-0.31%	-2.24%	-0.44%
	0.9	95.528	1.443	137.458	0.993	96.244	1.428	133.638	0.995	0.74%	-1.08%	-2.86%	0.20%
	1	100.713	1.254	141.632	0.890	100.330	1.261	137.319	0.886	-0.38%	0.54%	-3.14%	-0.43%
	1.1	105.698	1.089	145.626	0.797	104.443	1.097	140.617	0.793	-1.20%	0.71%	-3.56%	-0.41%
	1.2	110.429	0.947	149.418	0.712	108.499	0.956	143.276	0.719	-1.78%	0.96%	-4.29%	0.88%
	1.3	114.868	0.824	152.992	0.637	111.991	0.829	146.271	0.640	-2.57%	0.59%	-4.59%	0.52%
	1.4	118.989	0.718	156.337	0.570	115.537	0.716	149.197	0.563	-2.99%	-0.34%	-4.79%	-1.12%
	1.5	122.780	0.627	159.450	0.509	118.252	0.631	151.332	0.512	-3.83%	0.60%	-5.36%	0.56%
	1.6	126.240	0.548	162.331	0.456	121.263	0.543	153.474	0.459	-4.10%	-1.03%	-5.77%	0.61%
	1.7	129.378	0.481	164.984	0.408	123.167	0.485	155.752	0.407	-5.04%	0.78%	-5.93%	-0.17%
	1.8	132.209	0.423	167.417	0.365	125.342	0.423	157.465	0.365	-5.48%	0.13%	-6.32%	-0.12%
	1.9	134.752	0.373	169.642	0.328	126.811	0.378	158.905	0.330	-6.26%	1.56%	-6.76%	0.63%
	2	137.029	0.329	171.669	0.294	128.642	0.327	160.263	0.298	-6.52%	-0.59%	-7.12%	1.41%

approximated by an 80/20 *Pareto* distribution, where 80% of the orders are fulfilled by just 20% of the storage. Additionally, the new storage system, operating with one robot, should achieve a throughput of approximately 80 retrievals per hour.

Based on these requirements, the optimisation example aims to address the following questions:

- Is there a modification that enables such a performance?
- If yes, how does the system configuration look like?
- How much space and volume demand is required?

Table 11. Comparison of the results throughput ϑ and relocation probability w_{rel} analytical approach versus discrete event simulation for a grid size of 10 by 10 and a parameter $\alpha = 0.6$

		Analytical approach		Discrete event simulation		Estimation error	
		$\vartheta[1h]$	w_{rel}	$\vartheta[1h]$	w_{rel}	$\vartheta[1h]$	w_{rel}
stack height sh	1	226.593	0.000	222.389	0.000	-1.89%	0.00%
	2	207.490	0.248	202.172	0.250	-2.63%	0.76%
	3	191.636	0.449	186.107	0.450	-2.97%	0.13%
	4	178.179	0.621	173.037	0.619	-2.97%	-0.43%
	5	166.449	0.774	161.734	0.773	-2.92%	-0.13%
	6	156.079	0.912	152.111	0.906	-2.61%	-0.60%
	7	146.828	1.038	143.123	1.043	-2.59%	0.47%
	8	138.514	1.155	135.690	1.152	-2.08%	-0.30%
	9	130.999	1.265	128.726	1.260	-1.77%	-0.44%
	10	124.169	1.369	122.311	1.370	-1.52%	0.07%
	11	117.935	1.467	116.628	1.465	-1.12%	-0.10%
	12	112.222	1.560	111.187	1.561	-0.93%	0.09%
	13	106.968	1.649	106.633	1.645	-0.31%	-0.21%
	14	102.120	1.734	102.119	1.731	0.00%	-0.20%
	15	97.635	1.816	98.029	1.808	0.40%	-0.43%
	16	93.474	1.895	94.238	1.887	0.81%	-0.43%
	17	89.606	1.971	90.827	1.952	1.34%	-1.01%
	18	86.001	2.045	87.186	2.044	1.36%	-0.04%
	19	82.634	2.117	84.159	2.110	1.81%	-0.34%
	20	79.484	2.186	80.949	2.194	1.81%	0.35%
	21	76.532	2.254	78.247	2.255	2.19%	0.05%
	22	73.760	2.320	75.840	2.316	2.74%	-0.14%
	23	71.153	2.384	73.478	2.381	3.16%	-0.12%
	24	68.698	2.446	71.477	2.416	3.89%	-1.25%
	25	66.383	2.507	68.590	2.519	3.22%	0.46%

This example serves as an ideal use case for demonstrating the analytical approach and its intended application. The required storage systems are evaluated based on the geometric and throughput requirements outlined in Table 8. Given the vast number of conceivable system configurations, the increment for the number of stacks along the x- and z-axes is set to 5. Applying a tolerance of $\pm 2\%$ deviation in total storage capacity reduces the number of possible solutions to 7, making the analysis more manageable. The required computation time and power are minimal, as no algorithm is needed.

The results derived from the analytical approach are summarised in Table 9 sorted by the number of stacks. The space demand includes the area required for the storage system and an additional 2 m for the pre-zone with the picking station. The volume calculation also accounts for a two-meter gap above the storage grid to accommodate robot operation. Figure 11 illustrates four key and illustrative cases from Table 9.

As shown in Table 9, the maximum achievable throughput is approximately 98 picks per hour (case number 7). Generally, smaller grid sizes and higher container stacks result in lower throughput and reduced space demand. The table also reveals an interesting observation: the 20 by 20 grid provides better performance with a smaller footprint and volume requirement compared to the 40 by 10 system. This advantage is attributed to the positioning of the I/O shaft along the wider axis.

For instance, case number 4 (25 by 20 grid with $sh = 20$) achieves nearly 70 retrievals per hour and requires only 210 m^2 of space. Additionally, the volume required is comparable to that of the smallest system (case number 2). This configuration offers a favourable balance between high performance and compact footprint.

Table 12. Relocation probability w_{rel} depending on the stack height sh and the parameter α .

	α	0.001	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
sh	1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	2	0.333	0.332	0.318	0.303	0.289	0.275	0.261	0.248	0.235	0.223	0.211
	3	0.636	0.633	0.603	0.570	0.538	0.507	0.478	0.449	0.422	0.396	0.371
	4	0.919	0.915	0.866	0.813	0.762	0.713	0.666	0.621	0.579	0.538	0.500
	5	1.189	1.182	1.114	1.040	0.969	0.901	0.836	0.774	0.715	0.660	0.609
	6	1.448	1.439	1.350	1.254	1.162	1.074	0.991	0.912	0.837	0.768	0.703
	7	1.698	1.687	1.577	1.458	1.345	1.237	1.134	1.038	0.948	0.864	0.787
	8	1.942	1.929	1.796	1.654	1.519	1.390	1.269	1.155	1.050	0.952	0.862
	9	2.180	2.164	2.009	1.843	1.685	1.536	1.396	1.265	1.144	1.033	0.930
	10	2.412	2.394	2.216	2.026	1.846	1.675	1.516	1.369	1.232	1.107	0.993
	11	2.640	2.620	2.418	2.204	2.001	1.809	1.631	1.467	1.315	1.177	1.051
	12	2.864	2.841	2.616	2.377	2.151	1.939	1.741	1.560	1.393	1.242	1.106
	13	3.085	3.060	2.811	2.546	2.297	2.063	1.847	1.649	1.468	1.304	1.156
	14	3.303	3.275	3.001	2.712	2.439	2.184	1.949	1.734	1.538	1.362	1.204
	15	3.517	3.487	3.189	2.874	2.577	2.302	2.048	1.816	1.606	1.418	1.249
	16	3.729	3.696	3.373	3.033	2.713	2.416	2.143	1.895	1.671	1.470	1.292
	17	3.939	3.903	3.556	3.189	2.845	2.527	2.236	1.971	1.733	1.521	1.332
	18	4.146	4.108	3.735	3.342	2.975	2.636	2.326	2.045	1.793	1.569	1.371
	19	4.351	4.310	3.912	3.494	3.103	2.742	2.414	2.117	1.851	1.616	1.408
	20	4.554	4.511	4.088	3.643	3.228	2.846	2.499	2.186	1.907	1.660	1.443
	21	4.756	4.710	4.261	3.790	3.351	2.948	2.582	2.254	1.961	1.703	1.477
	22	4.955	4.907	4.432	3.934	3.472	3.048	2.664	2.320	2.014	1.745	1.510
	23	5.153	5.102	4.602	4.078	3.591	3.146	2.744	2.384	2.065	1.785	1.541
	24	5.350	5.296	4.770	4.219	3.709	3.242	2.822	2.446	2.114	1.824	1.572
	25	5.545	5.488	4.936	4.359	3.824	3.337	2.898	2.507	2.163	1.862	1.601
	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	
	1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	2	0.200	0.189	0.179	0.169	0.159	0.150	0.142	0.133	0.126	0.118	0.111
	3	0.347	0.324	0.303	0.283	0.264	0.246	0.229	0.213	0.198	0.184	0.171
	4	0.463	0.429	0.397	0.367	0.340	0.313	0.289	0.267	0.246	0.227	0.209
	5	0.560	0.515	0.473	0.434	0.398	0.365	0.334	0.306	0.280	0.256	0.234
	6	0.643	0.587	0.536	0.488	0.445	0.405	0.369	0.336	0.305	0.278	0.253
	7	0.715	0.649	0.589	0.534	0.484	0.438	0.397	0.359	0.325	0.295	0.267
	8	0.779	0.704	0.635	0.573	0.516	0.466	0.420	0.378	0.341	0.308	0.278
	9	0.837	0.753	0.676	0.607	0.545	0.489	0.439	0.394	0.354	0.319	0.287
	10	0.890	0.797	0.712	0.637	0.570	0.509	0.456	0.408	0.365	0.328	0.294
	11	0.938	0.836	0.745	0.664	0.592	0.527	0.470	0.420	0.375	0.335	0.300
	12	0.983	0.873	0.775	0.688	0.611	0.543	0.483	0.430	0.383	0.342	0.305
	13	1.024	0.907	0.803	0.710	0.629	0.557	0.494	0.439	0.390	0.347	0.310
	14	1.063	0.938	0.828	0.730	0.645	0.570	0.504	0.447	0.396	0.352	0.314
	15	1.100	0.967	0.851	0.749	0.660	0.582	0.513	0.454	0.402	0.357	0.317
	16	1.134	0.995	0.873	0.766	0.673	0.592	0.522	0.460	0.407	0.361	0.320
	17	1.166	1.020	0.893	0.782	0.686	0.602	0.529	0.466	0.411	0.364	0.323
	18	1.197	1.045	0.912	0.797	0.697	0.611	0.536	0.471	0.416	0.367	0.325
	19	1.226	1.068	0.930	0.811	0.708	0.619	0.542	0.476	0.419	0.370	0.327
	20	1.254	1.089	0.947	0.824	0.718	0.627	0.548	0.481	0.423	0.373	0.329
	21	1.281	1.110	0.963	0.836	0.727	0.634	0.554	0.485	0.426	0.375	0.331
	22	1.306	1.130	0.978	0.848	0.736	0.641	0.559	0.489	0.429	0.377	0.333
	23	1.330	1.149	0.993	0.859	0.745	0.647	0.564	0.492	0.431	0.379	0.334
	24	1.354	1.167	1.006	0.869	0.753	0.653	0.568	0.496	0.434	0.381	0.335
	25	1.376	1.184	1.019	0.879	0.760	0.659	0.572	0.499	0.436	0.383	0.337

7. Conclusion

The increasing strain on logistics and supply chains due to global crises, coupled with the rise in e-commerce, is accelerating the digitalisation and automation of warehousing. RCS/R systems are in high demand because they offer fully autonomous operation via robots, high storage density, modular scalability, and high availability. However, there are hardly any

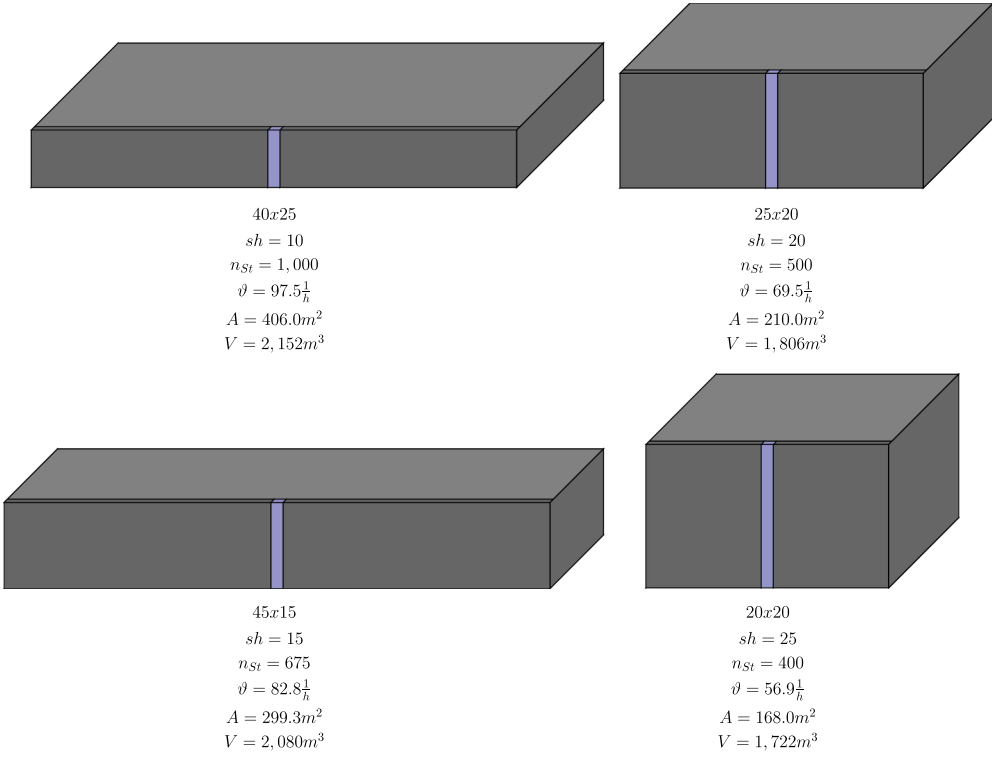


Figure 11.4 showcases of the application example.

models available that provide quick and accurate predictions of throughput for specific scenarios. Neither academic research nor system suppliers has offered straightforward performance approximations that focus on demand structure, which is crucial for RCS/R systems where performance is heavily influenced by stack height and the number of required relocations.

This paper introduces a novel analytical approach to determine the throughput of an RCS/RS with a single operating robot, considering any arbitrary article demand distribution over the stack height. The approach uses the *Pareto* distribution as an example, reflecting the common 80/20 distribution promoted by material handling providers for their storage systems.

The time-continuous model builds on a probabilistic CTM developed by Trost and Eder (2024b), with extensions to handle probability-distributed access structures. The approach has been validated against discrete event simulations, demonstrating high accuracy. This model is notable for its fast, straightforward calculations, easy applicability, and broad range of uses.

For a 54/20 *Pareto*-distributed access structure (where 54% of the demand can be met with just 20% of the stack height), the relocation probability is reduced by nearly half for a stack height of 10 containers, resulting in up to a 50% increase in throughput. Increasing the distribution parameter, i.e. rising inequality, while keeping stack height constant lowers the relocation probability and increases throughput. Conversely, a uniform distribution leads to decreased throughput as

stack height increases. The impact of the storage system's filling degree on performance under a *Pareto*-distributed demand is relatively minor.

This analytical model can benefit system suppliers, warehouse customers, and consultants by allowing the analysis of any demand distribution and predicting RCS/RS performance. The throughput of one robot can serve as an input parameter for existing performance models such as Zou et al. (2018) or Trost and Eder (2024a), which typically assume random access.

Future research could expand this approach to multiple operating robots and class-based storage strategies, incorporating different routing logics to enhance performance. Additionally, studies could develop a three-dimensional class-based zoning approach and investigate scenarios with multiple I/O shafts and picking stations along various grid edges. Comparative studies could then assess different RCS/RS types and directly compare various storage systems, such as AS/RS, SBS/RS, RCS/RS, or RMFS. Given recent trends in automated storage systems prioritising energy efficiency due to sustainability concerns, and the emphasis by RCS/RS providers on their high energy efficiency, future research could focus on developing analytical models to evaluate the energy consumption of RCS/RS.

Acknowledgments

Open access funding is provided by TU Wien (TUW). This work was supported by the TU Wien University Library through its Open Access Funding Program. The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Program.

Author contributions

CRediT: **Philipp Trost**: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing; **Michael Eder**: Supervision.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Author contributions statement

All authors contributed to the study's conception and design. Material preparation, data collection and analysis were performed by Philipp Trost and Michael Eder. The first draft of the manuscript was written by Philipp Trost and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Data availability statement

The datasets generated during and/or analysed during the current study are available from the corresponding author upon reasonable request. To enhance system understanding, two relevant links are provided below:

https://www.youtube.com/watch?v=9UDgSofF_AY

<https://www.youtube.com/watch?v=g0xKO7Jv68c>

References

- Autostore. (2023, January). Autostore. online. Retrieved from <https://de.autostoresystem.com/>
- Beckschaefer, M., Malberg, S., Tierney, K., & Weskamp, C. (2017). Simulating storage policies for an automated grid-based warehouse system [Proceedings]. *Computational Logistics: 8th International Conference, ICCL 2017*, Springer International Publishing, Southampton, UK, October, 2017.
- Brüggemann, H., & Bremer, P. (2020). *Grundlagen qualitätsmanagement*. Springer Fachmedien Wiesbaden.
- Chen, X., Yang, P., & Shao, Z. (2022, September). Simulation-based time-efficient and energy-efficient performance analysis of an overhead robotic compact storage and retrieval system. *Simulation Modelling Practice and Theory*, 119, 102560. <https://doi.org/10.1016/j.simpat.2022.102560>
- Eder, M. (2022, June). An analytical approach for a performance calculation of shuttle-based storage and retrieval systems with multiple-deep and class-based storage. *Production and Manufacturing Research*, 10(1), 321–336. <https://doi.org/10.1080/21693277.2022.2083715>
- Ekren, B., Sari, Z., & Lerher, T. (2015). Warehouse design under class-based storage policy of shuttle-based storage and retrieval system. *IFAC-Papersonline*, 48(3), 1152–1154. <https://doi.org/10.1016/j.ifacol.2015.06.239>
- Galka, S., & Scherbarth, C. (2021, September 15th-17th). Simulationsbasierte Untersuchung der Grenzproduktivität von Robotern in einem AutoStore-Lagersystem. In *Simulation in Produktion und Logistik*, Erlangen, Germany, Cuvillier Verlag.
- Hausman, W. H., Schwarz, L. B., & Graves, S. C. (1976). Optimal storage assignment in automatic warehousing systems. *Management Science*, 22(6), 629–638.
- Kartnig, G., Trost, P., & Eder, M. (2023, June 19th-23rd). RCS/RS under throughput investigation. Proceedings of the 16th International Material Handling Research Colloquium (IMHRC) and the 11th International Scientific Symposium on Logistics (ISSL), Dresden, Saxony, Germany. 1–8.
- Lehmann, T., & de Koster, M. R. (2024). Determining the throughput capacity of multi-deep storage systems. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4840510>
- Petersen, C. G., Aase, G. R., & Heiser, D. R. (2004, August). Improving order-picking performance through the implementation of class-based storage. *International Journal of Physical Distribution and Logistics Management*, 34(7), 534–544. <https://doi.org/10.1108/09600030410552230>
- Schenone, M., Mangano, G., Grimaldi, S., & Cagliano, A. C. (2020, January). An approach for computing as/r systems travel times in a class-based storage configuration. *Production and Manufacturing Research*, 8(1), 273–290. <https://doi.org/10.1080/21693277.2020.1781703>
- Schlittgen, R. (2008). *Einführung in die Statistik - Analyse und Modellierung von Daten*. De Gruyter GmbH, Walter.
- Tjeerdsma, S. (2019). *Redesign of the autostore order processing line, a multi-scenario discreteevent simulation study* [Unpublished master's thesis]. University of Twente.
- Trost, P., & Eder, M. (2024a, July). An analytical approach for the performance calculation of an rcs/rs with several picking stations. *The International Journal of Advanced Manufacturing Technology*, 134(1–2), 627–644. <https://doi.org/10.1007/s00170-024-14124-3>
- Trost, P., & Eder, M. (2024b, February). An analytical performance approach for rcs/rs with one robot serving multiple stack heights under a one-path relocation strategy. *Scientific Reports*, 14(1), <https://doi.org/10.1038/s41598-024-53884-6>
- Trost, P., & Eder, M. (2024c, March). A performance calculation approach for a robotic compact storage and retrieval system (rcs/rs) serving one picking station. *Production and Manufacturing Research*, 12(1). <https://doi.org/10.1080/21693277.2024.2336056>
- Trost, P., Karting, G., & Eder, M. (2023). Simulation study of rcs/r-systems with several robots serving one picking station. *FME Transactions*, 51(2), 201–210. <https://doi.org/10.5937/fme2302201T>

- Tutam, M., Liu, J., & White, J. A. (2024, August). Consideration of skewness in designing robotic compact storage and retrieval systems. *Expert Systems with Applications*, 248, 123361. <https://doi.org/10.1016/j.eswa.2024.123361>
- Wang, R., Yang, P., Gong, Y., & Chen, C. (2023, December). Operational policies and performance analysis for overhead robotic compact warehousing systems with bin reshuffling. *International Journal of Production Research*, 62(14), 1–16. <https://doi.org/10.1080/00207543.2023.2289643>
- Xu, X., Zhao, X., Zou, B., & Li, M. (2018, December). Optimal dimensions for multi-deep storage systems under class-based storage policies. *Cluster Computing*, 22(3), 861–875. <https://doi.org/10.1007/s10586-018-2873-9>
- Yu, Y., & de Koster, M. (2008, December). Designing an optimal turnover-based storage rack for a 3d compact automated storage and retrieval system. *International Journal of Production Research*, 47(6), 1551–1571. <https://doi.org/10.1080/00207540701576346>
- Yu, Y., de Koster, R. B., & Guo, X. (2015, January). Class-based storage with a finite number of items: Using more classes is not always better. *Production and Operations Management*, 24(8), 1235–1247. <https://doi.org/10.1111/poms.12334>
- Zaerpour, N., de Koster, R. B., & Yu, Y. (2013, May). Storage policies and optimal shape of a storage system. *International Journal of Production Research*, 51(23–24), 6891–6899. <https://doi.org/10.1080/00207543.2013.774502>
- Zou, B., de Koster, R., & Xu, X. (2018, August). Operating policies in robotic compact storage and retrieval systems. *Transportation Science*, 52(4), 788–811. <https://doi.org/10.1287/trsc.2017.0786>