logistics

IMPACT FACTOR 4.4

CITESCORE 8.1

Article

# Multi-Agent Path Planning for a Multi-Deep Four-Way Shuttle-Based System

Giacomo Lupi, Andrea L'Afflitto, Riccardo Manzini and Gabriele Sirri

MDPI

https://doi.org/10.3390/logistics10070155

logistics

Article

# Multi-Agent Path Planning for a Multi-Deep Four-Way Shuttle-Based System

**Giacomo Lupi [1], Andrea L'Afflitto [2], Riccardo Manzini [1,*] and Gabriele Sirri [1]**

[1] Department of Industrial Engineering, University of Bologna, Viale del Risorgimento 2, 40136 Bologna, Italy; giacomo.lupi2@unibo.it (G.L.); gabriele.sirri4@unibo.it (G.S.)
[2] Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA 24060, USA; a.lafflitto@vt.edu
* Correspondence: riccardo.manzini@unibo.it

**Abstract**

*Background*: Four-way shuttle-based storage and retrieval systems (FSS/RSs) have recently emerged as flexible and scalable solutions for high-density warehousing, enabling shuttle movement in four directions and supporting multi-deep dual-access storage configurations. However, these features increase the complexity of vehicle coordination and collision management. This study proposes a multi-agent path-planning methodology for FSS/RSs with multi-deep dual-access lanes hosting homogeneous items. *Methods*: An A*-based path-planning framework was developed and integrated with a dynamic collision-management strategy comprising collision detection, priority assignment, and collision avoidance. The methodology was evaluated through a multi-scenario analysis considering different fleet sizes, priority-assignment strategies, safety-area extensions, transaction-entry patterns, and collision-management policies. *Results*: The results show that fleet size is the most influential operational parameter, significantly affecting throughput, waiting times, and collision frequency. Increasing the number of vehicles improves system productivity but also increases traffic interactions and congestion. The analyses further highlight the effects of safety-area size, priority rules, and transaction-entry patterns on operational performance and system robustness. *Conclusions*: The proposed methodology effectively combines path planning and collision management in four-way shuttle systems, providing a decision-support tool for evaluating operational trade-offs among throughput, congestion control, and system stability in multi-deep dual-access warehouse environments.

**Keywords:** four-way shuttle systems; path planning; A* algorithm; collision management; multi-deep lane; automated material handling



Academic Editor: Robert Handfield

Received: 12 May 2026
Revised: 29 June 2026
Accepted: 6 July 2026
Published: 9 July 2026



## 1. Introduction

Shuttle-based storage and retrieval systems (SBS/RSs) have emerged as highly flexible and scalable solutions for automated material handling and warehousing in modern production and distribution centers. These automated systems consist of autonomous vehicles operating at multiple levels within a rack system, performing storage and retrieval (S/R) transactions without human intervention. Over the past two decades, shuttle-based technologies have significantly evolved, enabling enhanced efficiency and guaranteeing high levels of space utilization and throughput in various sectors.

A key feature of SBS/RSs is their ability to move both horizontally and vertically, providing access to various storage locations. This improves the overall efficiency of the S/R processes by reducing travel time and increasing throughput. Previous studies have demonstrated the superior performance of shuttle-based systems in terms of space utilization and system throughput compared with traditional automated storage solutions [1–4].

Despite the initial investment costs, the long-term benefits of shuttle-based systems, including labor cost reduction, improved order accuracy, and enhanced space utilization, have driven widespread adoption. As these systems evolve, ongoing research focuses on improving energy efficiency, optimizing control strategies, and ensuring seamless integration with other automation and robotic technologies.

SBS/RSs and automated vehicle S/R systems (AVS/RSs) with homogeneous multi-deep lanes are traditionally equipped with shuttles moving along the main cross-aisle and satellites moving through the lanes, as described by [3,4]. Recent innovations have enhanced shuttle-based systems. The introduction of four-way shuttle S/R systems (FSS/RSs) allows shuttles to move along the main cross-aisle and within the lanes, increasing system flexibility and operational efficiency compared with traditional systems.

This multidirectional capability allows vehicles to cover an entire warehouse more quickly and efficiently, thereby reducing bottlenecks and improving response times. Figure 1 provides a three-dimensional view of the multi-deep FSS/RS. The system comprises multiple tiers, each structured as a horizontal ($z$-axis) and longitudinal ($x$-axis) aisle grid. These aisles allow shuttles to move freely in four directions (forward, backward, left, and right), thereby enabling them to navigate the storage area dynamically. The storage lanes are composed of individual storage locations where unit loads (ULs), represented as blue cubes, are stored and retrieved according to the last in first out (LIFO) strategy until the lane depth (k in Figure 1) is reached. Unlike traditional systems, where lane access is restricted to dedicated vehicles (i.e., satellites), a key feature of the FSS/RS is that the generic shuttle can access lanes directly from the main cross-aisle, enabling them to store or retrieve ULs. The lifts positioned at the corners of the system provide vertical connectivity between the tiers. The entire system is governed by a three-dimensional coordinate framework, where the $x$-axis defines the movement along the main cross-aisles, the $y$-axis corresponds to the vertical connection between tiers, and the $z$-axis represents access to the storage lanes and a set of auxiliary drive-through aisles. In this study, the term vehicle is consistently used to refer to a four-way shuttle, i.e., the handling unit responsible for storage and retrieval operations within the system. This ensures clarity when discussing vehicle movements, empty repositioning, and lane allocation. When a shuttle is empty, it is allowed to move both along the main-cross aisle and the drive-through auxiliary aisles and to enter the storage lanes even if unit loads are already stored inside. However, direct movement between adjacent lanes is not permitted; lane access is only possible from the aisles. If a shuttle is loaded, it can move between different main cross-aisles or in a storage lane only to perform a S/R transaction. This integrated design allows highly dynamic operations. The ULs are vertically transported by lifts, and shuttles access storage lanes via the main cross-aisles.

Another significant advantage of the FSS/RS is its scalability. System productivity can be effectively enhanced by adjusting the size of the shuttle fleet. In traditional systems, productivity improvements are limited to adding a satellite to each level, which yields only marginal benefits, as demonstrated by [3].

These systems have been increasingly adopted in industries that require high-speed and flexible operations, such as e-commerce, pharmaceuticals, and spare-parts distribution. Moreover, decentralizing control, with multiple shuttles operating simultaneously, increases system resilience and reduces the impact of potential shuttle downtimes. Furthermore, intelligent control algorithms, real-time data analytics, and integration with warehouse-management systems (WMS) enable FSS/RS to be more efficient and responsive to the needs of modern supply chains.

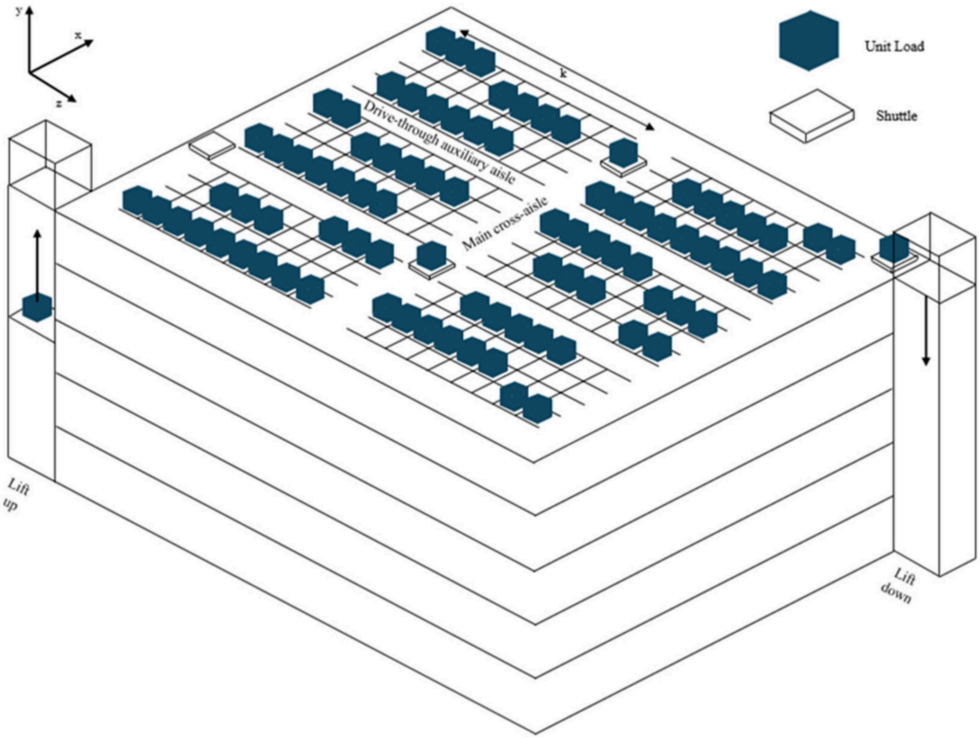


**Figure 1.** Three-dimensional view of a four-way shuttle system.

A significant advantage of four-way shuttle systems is their ability to support dual-access multi-deep lanes. The dual-access feature significantly enhances the flexibility and efficiency of S/R operations, particularly in homogeneous lanes. Homogeneity means that the generic storage lane hosts ULs for the same item. In a traditional shuttle-based S/R system, the lane is single access, meaning that only one type of UL (e.g., a single SKU or a single production batch) can be stored in a lane, and retrieval must occur from one fixed side (i.e., the main cross-aisle in Figure 1) based on LIFO movements. With dual-access lanes enabled by four-way shuttles, two different types of UL (i.e., homogenous items) can be stored within the same lane and are accessible from both ends. This allows for dual entry and exit points, thereby providing multiple benefits. The primary advantage of dual-access lanes is the increased storage density without compromising operational performance. Because two different ULs can be stored in a single lane, the warehouse can accommodate more product varieties in a given area. Four-way shuttles can retrieve ULs from either side of a lane, allowing dynamic UL retrieval depending on which exit is closer or less congested at any given time. This reduces the travel distances for shuttles, minimizing bottlenecks and vehicle congestion, especially in high-traffic operating time windows.

Managing different ULs within the same lane also improves the operational efficiency. Selecting the UL type that can be stored in a specific lane is the so-called assignment strategy (AS) [4]. Instead of dedicating an entire lane to a single UL, four-way shuttle systems can use space more dynamically by adjusting the inventory positioning based on real-time demands. This feature is particularly valuable in production systems with a wide variety of products or fluctuating demand, as it allows for the rapid reconfiguration of storage assignments without system and layout changes.

Figure 2 shows an example of a multi-deep dual-access FSS/RS tier layout. Three main cross-aisles are observed along the *x*-axis. The *z*-axis comprises dual-access storage lanes and four auxiliary aisles. Each lane has a lane depth (LD) k, which is equal to 12 in Figure 2. Access is enabled from both sides where the main cross-aisles are located, allowing shuttles to retrieve or store ULs from either end. Because of dual access, a lane can be divided into two semi-lanes (SLs) with depths k1 and k2, each capable of hosting homogeneous items. An item is the identifier of a cluster of ULs and the same color in Figure 2 represents homogeneous ULs. In these systems, the lane depth varies according to depth k1 and k2 can vary dynamically.

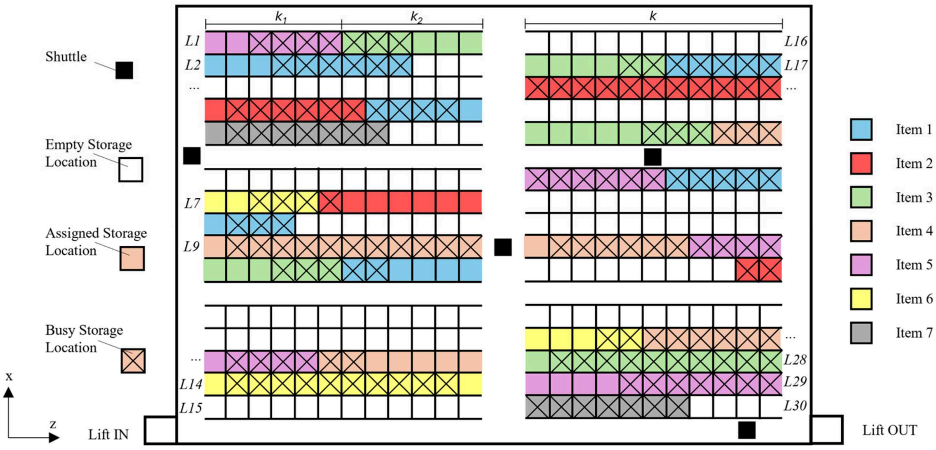


**Figure 2.** Example of four-way system tier layout.

A generic storage location can be empty, busy, or assigned to a cluster of items. A lane/SL is empty if all its storage locations are empty, but is busy if at least one storage location hosts a UL. An assigned lane/SL hosts items belonging to a homogeneous cluster. If a lane/SL is busy, it is assigned to the item. When the last UL exits the SL, the SL becomes empty and re-assignable. That is, the SL becomes available to host new items. When the two-sided SLs of the same lane are empty, the k-capacity lane becomes available to host a new item or be divided into two new SLs that differ from the previous ones.

Figure 2 exemplifies some configurations of items assigned to lanes and SLs:

- Lane L1 is divided into two SLs. The left SL has depth k1 = 6 and is assigned to Item 5; the right SL has depth k2 = 6 and is assigned to Item 3.
- Lane L2 is divided into two SLs. The left SL has depth k1 = 9 and is assigned to Item 1; the right SL has depth k2 = 3 and is empty.
- Lane L9 has depth k = 12 and is assigned to Item 4.
- Lane L15 has depth k = 12 and is empty.

The generic shuttle shown in Figure 2 is the vehicle responsible for handling the ULs, and it can move along the main aisles and enter the lanes from both sides. A shuttle loading a UL can move through the main and auxiliary aisles or empty lanes. If the shuttle is empty, it can also be driven through the busy lanes hosting unit loads. For a storage transaction, the shuttle brings the UL from the Lift IN to its storage location, whereas for a retrieval transaction, the shuttle brings the UL from its storage location to the Lift OUT. For a retrieval transaction, the shuttle can exit the lane only from the side where no ULs block its travel. For example, in lane L7 in Figure 2, the retrieval transaction for Item 2 (i.e., the red one) can exit only from the right side of the lane, because ULs of Item 6 (i.e., the yellow ones) are on the left side.

Multi-agent path planning in four-way shuttle systems is crucial for maximizing the system throughput (ST) performance, which refers to the rate at which items are stored and retrieved. Throughput also directly impacts the overall efficiency of warehouse operations, as a higher throughput means that more orders can be processed in a given time, thereby reducing lead times and improving productivity.

The movement of shuttles within the system along the x- and z-axes must be carefully managed to avoid congestion and minimize travel times. Efficient routing algorithms help determine the most effective paths for shuttles, reduce unnecessary movement, and ensure that items are stored or retrieved within the shortest cycle time. Poor-quality path planning can lead to bottlenecks, where multiple shuttles occupy the same lane or aisle, resulting in delays, congestion, and decreased throughput.

By optimizing the shuttle paths, the system can handle more S/R transactions simultaneously, thereby increasing the throughput and maximizing the utilization of the nominal storage capacity of the system. This is particularly important in high-demand environments, where even small inefficiencies in shuttle routing can significantly affect the overall performance.

This study focuses on developing a methodology that adopts a path planning algorithm to determine the most efficient routes for multiple fleet shuttles while also handling potential collisions to enhance system throughput. The system object of this study is an FSS/RS with tier-captive shuttles, where each shuttle is confined to a specific tier. The system contains multi-deep lanes that can host up to two different items in a single lane, and the depth of each semi-lane is dynamically time-variable. The path planning process is based on the A* algorithm, which is widely recognized for its ability to find optimal routes by considering both distance and obstacles in route finding. The proposed methodology ensures shuttles move along efficient paths, minimizing delays and improving the system's overall throughput.

The novelty of this study lies in the exploration of path-planning strategies in an FSS/RS that adopt a multi-deep lane with dual access. In particular, this study aims to understand how different leverages in the system designer and user domains influence the material-handling system performance. This perspective shifts the discussion from proving optimality to analyzing the impact of strategic decision-making in vehicle coordination, providing new insights into the trade-offs among efficiency, congestion management, and operational robustness in this specific context, which has not yet been explored.

The literature has a notable gap regarding the application of pathfinding algorithms to four-way shuttle systems with multi-deep configurations. The existing research has predominantly focused on systems with single-deep lanes. By contrast, this study examines a scenario in which each lane accommodates up to two items while maintaining homogeneity within the semi-lane. These deep-lane systems are increasingly used in industrial applications, accounting for more than 20 locations in some production systems, such as in the food and beverage industry.

Additionally, the issue of collision management in such environments has not been consistently deepened. This study addresses this critical gap by implementing an improved A* algorithm to generate collision-free paths, which enhances these material handling systems' efficiency and operational reliability.

The proposed methodology is applied to a numerical example inspired by a realistic system. A multi-scenario comparative and competitive analysis is conducted. The adopted analysis approach allows monitoring the impact of various leverages on the system performance, such as the number of vehicles, providing insights into how different factors influence throughput and efficiency under different operational conditions.

## 2. Related Works

To better contextualize the proposed study, the last part of this section provides a comprehensive literature review of the key concepts and methodologies underlying SBS/RSs

https://doi.org/10.3390/logistics10070155

and FSS/RSs. Given the growing complexity and demand for automated-storage solutions, this literature review explores the relevant studies and technological advancements in three main areas of research.

The first addresses the foundational elements and evolution of shuttle-based S/R systems by examining their operational characteristics, performance evaluations, and advantages in high-density storage environments. This overview establishes a context for understanding the enhancements owing to four-way mobility. Numerous studies have analyzed the performance of SBS/RSs using both analytical and simulation methodologies. These approaches are pivotal for evaluating system efficiency, throughput, and response time across different configurations and operational scenarios. Together, these methods offer a robust framework for understanding the capabilities and limitations of SBS/RSs. Ref. [5] evaluated the best rack configuration for an SBS/RS under various class-based storage policies. They used a simulation to model the problem and evaluated the lift and shuttle utilization performances and S/R cycle times. Ref. [6] explored the design of an optimal rack for multi-deep AS/RS development models and solving methods. In their system, each UL in any storage location can be accessed individually by combining an automated S/R machine and multiple independently operating automated depth-movement mechanisms. Ref. [7] developed a software tool based on open queuing networks to support the evaluation of alternative SBS/RS configurations. The proposed approach enables the estimation of several operational indicators, including transaction cycle and waiting times, energy consumption and regeneration, queue lengths at lifts and shuttles, and the utilization levels of the system resources. Ref. [8] considered an SBS/RS in which a shuttle could perform both horizontal and vertical movements. They proposed analytical models to calculate cycle times for single- and dual-command cycles. Furthermore, a simulation model was developed to evaluate the analytical travel-time models. Ref. [9] developed a travel-time model for an SBS/RS under an alternative assignment policy and dispatching rule. They explored random class-based storage policies, and random distance- and demand-rate-based dispatching rules.

SBS/RS is often used for handling totes with shallow lanes, while AVS/RS is a class of SBS/RS that uses palletized ULs combined with deep-lane configurations. Given alternative layout configurations, Ref. [10] developed original analytic models for determining the traveled distance and time for single and dual command cycles. Their goal was to present guidelines for configuring the system layout, including determining the optimal shape ratio and the depth of the lanes. Ref. [11] introduced an analytical framework for multi-deep AVS/RSs and assessed its performance under various storage layouts, operating cycles, and transaction-assignment strategies. Ref. [3] analyzed the impact of storage policies and dispatching rules on system performance regarding space efficiency. This study aimed to support the decision-making process on the design, management, and control of an AVS/RS. Ref. [4] introduced and applied an original data-driven comparative multi-scenario methodology to measure and control AVS/RS performance. The approach was based on a hybrid analytical-simulative model and was applied to a real-world case study. This methodology considered various leverages (i.e., lane depth, assignment strategy, and dispatching strategy) and measured their impact on storage capacity, space efficiency, and system throughput.

Compared to traditional automated storage and retrieval systems (AS/RS), shuttle-based S/R systems demonstrate enhanced flexibility and scalability, which are essential for modern, high-throughput environments [1,2,4]. The shuttle-based design allows for independent, parallel movement of multiple shuttles, which reduces travel times and increases system throughput. These characteristics make shuttle-based particularly well-suited for

https://doi.org/10.3390/logistics10070155

applications requiring high storage density and fast retrieval cycles, thus substantially improving operational efficiency over conventional systems.

The second part of this literature review focuses on the application of pathfinding algorithms to vehicle routing for autonomous mobile robot (AMR) fleets. An AMR is capable of navigating and performing tasks in dynamic environments without human intervention. Pathfinding algorithms are critical for optimizing vehicle routes in automated S/R environments, where efficient and collision-free navigation is essential. The literature explores how these algorithms are adapted to handle the unique challenges posed by multivehicle coordination, dynamic environments, and real-time route recalculations to ensure optimal system throughput and operational efficiency. Ref. [12] identified and classified studies on the planning and control of AMRs in intralogistics. They provided an extended literature review highlighting how technological advances in AMR affect planning and control decisions. Choosing between centralized and decentralized controls for AMRs is crucial. Centralized control uses global information to optimize small and simple systems, whereas decentralized control, which relies on local information, is better suited for complex and large-scale systems with multiple objectives. Decentralization reduces the computation time and allows for quicker recovery from failures by distributing decision-making across robots. Determining the optimal level of decentralization is essential for strategic decisions such as scheduling, zoning, and path planning. Ref. [13] introduced and analyzed a cognitive industrial entity called Context-Aware Cloud Robotics. This cloud-based decision-making engine performs decentralized navigation (i.e., map processing), which can be shared among AMRs. The study highlighted that using simple AMRs and outsourcing decision-making to the cloud can reduce overall costs. Simultaneously, simulation modeling based on the status and location of AMRs can enhance energy efficiency. Researchers have used decomposition methods [14] and mathematical and statistical models [15] to examine and resolve the interactions between conflict-free vehicle routing and scheduling policies, focusing on their effects on production delays.

Additional studies addressed scheduling performance by analyzing indicators such as makespan, cycle-time variability, and vehicle earliness and tardiness. These works adopted multi-stage solution procedures that first reduced the search space and then identified high-quality scheduling solutions [16,17]. Ref. [18] discussed graph representations of the environment and graph search algorithms for a single AMR. Their research emphasized that the A* and D* Lite algorithms, modifications of Dijkstra's algorithm, are particularly effective. Ref. [19] stated that current simulation methods inadequately replicate AMR paths and behaviors in dynamic environments. In such settings, moving obstacles can temporarily obstruct the AMR's path. Their study presented experiments demonstrating the AMR's motion planning response to avoid these obstacles. They propose protocols designed to enhance the accuracy and quality of path planning simulations in dynamic environments.

In multi-vehicle intralogistics and warehousing environments, the geometrically shortest route is not necessarily the fastest because traffic interactions, congestion, and deadlock situations may occur. To address these challenges, several authors proposed mathematical models aimed at generating conflict-free vehicle movements [20–22]. Ref. [23] enhanced the Dijkstra algorithm to define initial transportation routes and identify potential congestion by evaluating vehicle movements within predefined time windows, allowing alternative routes to be selected when necessary. Ref. [24] analyzed both centralized and decentralized coordination approaches for managing shared resources. Their results showed that centralized coordination can improve throughput while reducing delays associated with vehicle interactions.

Recent research has also explored machine-learning techniques for routing and decision-making in automated warehouse systems. Ref. [25] proposed a reinforcement-learning-based hyper-heuristic to address large-scale AGV task-assignment and routing problems with the objective of reducing travel times. Ref. [26] developed a real-time reinforcement-learning framework capable of adapting routing decisions according to current operating conditions within smart warehouses. Ref. [27] compared single-agent and multi-agent learning approaches, highlighting the effectiveness of Proximal Policy Optimization (PPO) in improving overall manufacturing-system performance under decentralized decision-making environments.

Although significant advancements have been made in optimizing AMR path planning, coordination strategies, and decentralized control algorithms, dynamic environments and complex traffic patterns within automated warehousing systems, particularly four-way S/R systems have not been effectively addressed. This study aims to bridge these gaps by developing approaches that improve negotiation protocols and optimize vehicle routing under varying conditions, ultimately contributing to more efficient and reliable automated warehouse operations.

The third research area discussed in the literature is four-way shuttle technology, a recent development aimed at increasing the flexibility and efficiency within automated storage systems. By allowing multidirectional movement, four-way shuttles provide increased adaptability in high-density storage layouts, minimizing travel times (i.e., system throughput) and optimizing space utilization (i.e., efficiency). The literature on four-way shuttle systems is recent and primarily focuses on task scheduling using simulation-based approaches.

Ref. [28] proposed a control system to balance the number of shuttles in tier-to-tier FSS. They studied a solution that dynamically controlled each shuttle and load position, showing better performance in terms of throughput and utilization than a non-free balanced solution. Ref. [29] introduced a scheduling model based on a genetic algorithm to minimize the travel time of outbound orders and improve system transportation-scheduling efficiency. Their study demonstrated that scheduling S/R tasks according to optimal paths can significantly improve the operational efficiency of the system. Ref. [30] developed an automatic 3D modeling system for FSS with user-defined parameters based on the commercial software platform Flexsim. The system makes modeling more efficient and convenient, meeting the needs of planners and designers in real applications. Ref. [31] developed a scheduling method based on the Grey Wolf algorithm for FSS/RSs. This algorithm optimized the determination of the UL location allocation and delivery-vehicle scheduling. The results indicated that the optimization algorithm increased shelf utilization and reduced time consumption. Ref. [32] used a parallel operation strategy to reduce resource wastage and achieve the sustainable development of a single-deep four-way shuttle system. They used queuing network models verified by a commercial software platform, Arena, demonstrating an average reduction of 12.6% in the system-response time compared with existing methods. Ref. [33] established a mathematical model for scheduling optimization based on the minimum time for S/R transactions and path optimization for an FSS/RS. They used an improved genetic algorithm to solve task planning and an improved A* algorithm to solve the path-optimization problem. To manage the potential collisions of vehicles, they assumed that only one shuttle could operate simultaneously in the same section of the aisle. Ref. [34] proposed an improved genetic algorithm combining a critical path-based neighborhood searching strategy which can ensure the effectiveness of local search on AGVs. The results obtained by the algorithm showed significant advantages, proving the effectiveness of the proposed method and critical path-searching strategy. Ref. [35] analyzed the inbound operation process of an FSS/RS by introducing a flexible flow-shop-scheduling model to address the scheduling problem in the system.

https://doi.org/10.3390/logistics10070155

Additionally, they proposed an improved genetic algorithm based on double-layer encoding to solve this problem. They compared the results obtained using a traditional genetic algorithm and an ant-colony algorithm, demonstrating the superior efficiency and accuracy of their approach. Ref. [36] developed a task-scheduling model for a four-way single-deep tier-to-tier SBS/RS. They proposed an improved genetic algorithm that was validated using a practical case study to solve the model. Ref. [37] investigated the scheduling problem in a tier-captive FSS/RS equipped with multiple lifts and developed a two-stage heuristic solution approach. Their methodology combined an enhanced A* routing algorithm with a multi-phase conflict-management procedure to obtain collision-free shuttle movements. Computational experiments demonstrated the effectiveness of the proposed approach, achieving shorter makespan values than the alternative methods considered in the study. Ref. [38] studied an integrated scheduling optimization framework on multi-deep tier-to-tier four-way shuttle S/R systems. Their framework addressed request sequencing, equipment selection, and conflict-free path planning. They developed a control policy that selects the shuttle and lift with the earliest handover time, and demonstrated through large-scale experiments that their method improves operational efficiency by over 20% compared to traditional control strategies. Ref. [39] proposed a hierarchical graph-based approach for cooperative path planning in four-way shuttle-based SBS/RSs, dynamically adjusting edge properties to inherently eliminate congestion and deadlocks in large-scale environments. Ref. [40] developed an optimization model for four-way shuttle-based AS/RS, combining scheduling and collision-free path planning. Their study introduced a mathematical model for multi-type task operations aimed at minimizing total task completion time, solved via an improved genetic algorithm, and complemented with a dynamic scheduling strategy to reduce unnecessary tier-changing and improve overall efficiency. Ref. [41] investigated task allocation in four-way shuttle storage and retrieval systems. They modeled the problem as a partially observable Markov decision process and applied a deep reinforcement learning approach using an independent deep Q-network (IDQN) framework, demonstrating improved task completion times compared to auction and genetic algorithms.

Table 1 summarizes the most significant contributions from the literature according to the previously illustrated critical features. The fields in the comparative table are Authors, System Type, Lane depth (LD) describing the type (single, multiple, and variable) of lane depth adopted in the system, Modelling Approach, which refers to the adopted methodology for problem-solving, Collision Management, in case of the adoption of strategies to detect and solve collisions. Finally, performance refers to the following options/indicators/measures: service time (ST), waiting time (WT), collision detection (CD), and idle time (IT).

Despite significant research on SBS/RS and the advancements in multidirectional shuttle technology, a notable gap persists in studies that specifically integrate four-way shuttle dynamics with optimized path planning algorithms. Existing literature often treats shuttle-based S/R systems and path planning algorithms, such as A*, in isolation or with limited cross-application, focusing either on conventional two-way shuttles or single-vehicle route optimization. However, there is a limited exploration of how these algorithms, traditionally used in AGV and LGV fleets, can be tailored to maximize the efficiency of a fleet of four-way shuttles in complex, high-density storage environments. In addition, most prior studies primarily examine multi-deep or homogeneous lane structures, whereas there is little emphasis on systems with semi-lanes of variable depth. In contrast, in this study, each lane can store up to two items while maintaining homogeneity (i.e., assignment strategy) within the semi-lane. Moreover, the collision management challenge in such environments is not always explored.

**Table 1.** Literature Review.

| Authors | System Type | LD | Modeling Approach | Collision Management | Performance | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | ST | WT | CD | IT |
| Ekren et al. (2015) [5] | SBS/RS | Multiple | Simulative | No | Yes | No | No | No |
| Ekren & Akpunar (2021) [7] | SBS/RS | Multiple | Queueing Network | No | Yes | Yes | No | No |
| Lerher et al. (2021) [8] | SBS/RS | Multiple | Analytical-Simulative | No | Yes | No | No | No |
| Manzini et al. (2016) [10] | SBS/RS | Multiple | Analytical | No | Yes | No | No | No |
| D'Antonio & Chiabert (2019) [11] | SBS/RS | Multiple | Analytical | No | Yes | No | No | No |
| Battarra et al. (2022) [3] | SBS/RS | Multiple | Analytical-Simulative | No | Yes | No | No | No |
| Lupi et al. (2024) [4] | SBS/RS | Variable | Analytical-Simulative | No | Yes | Yes | No | No |
| Wan et al. (2017) [13] | AMR | Single | Queueing Network | No | Yes | No | No | No |
| Corréaet al. (2007) [14] | AGV | / | Mixed Integer Programming | Yes | Yes | No | Yes | No |
| Ghasemzadeh et al. (2019) [15] | AGV | / | Scheduling Algorithm | Yes | Yes | No | Yes | No |
| Fazlollahtabar et al. (2015) [16] | AGV | / | Mixed Integer Programming | Yes | Yes | No | Yes | No |
| Bakshi et al. (2019) [17] | AMR | / | Scheduling Algorithm | No | Yes | No | No | No |
| Liaqat et al. (2019) [19] | AMR | / | Simulative | No | Yes | No | No | No |
| Wu & Zhou (2007) [20] | AGV | / | Petri Nets | Yes | Yes | No | Yes | No |
| Saidi-Mehrabad et al. (2015) [21] | AGV | / | Ant Colony Optimization | Yes | Yes | No | Yes | No |
| Yang et al. (2018) [22] | AGV | / | Mixed Integer Programming | Yes | Yes | No | Yes | No |
| Zhang et al. (2018) [23] | AGV | / | Dijkstra Algorithm | Yes | Yes | No | Yes | No |
| Digani et al. (2019) [24] | AGV | / | Quadratic Optimization | Yes | Yes | No | Yes | No |
| Li et al. (2024) [25] | AGV | / | Reinforcement Learning | Yes | Yes | Yes | Yes | Yes |
| Ho et al. (2025) [26] | AGV | / | Reinforcement Learning | Yes | Yes | Yes | Yes | Yes |
| Ha & Chae (2018) [28] | FSS | Single | Analytical-Simulative | Yes | Yes | No | Yes | No |
| Song et al. (2020) [29] | FSS | Single | Genetic Algorithm | No | Yes | No | No | No |
| Cheng et al. (2020) [30] | FSS | Single | Simulative | No | Yes | No | No | No |
| Li et al. (2022) [31] | FSS/RS | Single | Grey Wolf Optimization | No | Yes | No | No | No |
| Mao et al. (2023a) [32] | FSS/RS | Single | Queuing Network | No | Yes | No | No | No |
| Mao et al. (2023b) [33] | FSS/RS | Single | A* Algorithm | Yes | Yes | No | Yes | No |
| Wu et al. (2024) [35] | FSS/RS | Single | Genetic Algorithm | No | Yes | No | No | No |
| Liu et al. (2023) [34] | FSS/RS | Single | Genetic Algorithm | No | Yes | No | No | Yes |
| Ren et al. (2024) [37] | FSS/RS | Single | A* Algorithm | Yes | Yes | No | Yes | No |
| Ren et al. (2025) [38] | FSS/RS | Multiple | Meta-heuristic algorithm | Yes | Yes | Yes | Yes | No |
| Han et al. (2024) [39] | FSS/RS | Multiple | Graph-based | Yes | Yes | Yes | Yes | Yes |
| Ding et al. (2024) [40] | FSS/RS | Multiple | Genetic Algorithm | Yes | Yes | Yes | Yes | Yes |
| Zhang et al. (2025) [41] | FSS/RS | Multiple | Deep reinforcement learning | Yes | Yes | Yes | Yes | Yes |
| Present study | FSS/RS | Multiple-Variable | A* Algorithm | Yes | Yes | Yes | Yes | Yes |

The originality of this work lies in the methodology developed and proposed for Four-Way Shuttle-Based Storage and Retrieval Systems. The methodology integrates A* path planning with a dynamic collision management framework that combines detection and resolution mechanisms specifically tailored to the operational characteristics of four-way shuttles. This enables the system to continuously detect conflicts, assign priorities, and recalculate trajectories in real time, ensuring efficient collision-free operations even in dense storage environments.

A further distinctive feature of the methodology is the explicit consideration of multi-deep storage, dual access, and double-item lanes, together with a homogeneous assignment strategy designed to minimize relocation and housekeeping movements that would otherwise reduce system efficiency.

In addition, the study conducts a multi-scenario analysis to evaluate the effects of various operational levers, including the number of vehicles, the depth of the safety area, the priority assignment strategy, the collision detection mechanism, and the S/R entry rates. By systematically varying these parameters using real industrial data, the analysis provides concrete insights into how each factor influences throughput and congestion. This data-driven approach addresses a significant gap in the literature, where systematic performance assessments of four-way systems remain scarce.

Through this methodological contribution, the paper differs from existing studies by integrating path planning, collision management, and system-specific features such as lane depth and assignment strategy into a unified framework. This advances the theoretical modeling of shuttle-based storage and retrieval systems and offers a practical decision-support tool to guide industrial applications in controlling and improving the performance of four-way shuttle systems.

The remainder of this paper is organized as follows. Section 3 describes the materials and methods adopted for path planning in a tier-captive four-way shuttle system. Section 4 presents the results of the numerical analyses based on a realistic four-way shuttle system. Section 5 discusses the main findings and managerial implications of the study. Finally, Section 6 provides the conclusions and outlines directions for future research.

## 3. Materials and Methods

This section introduces a methodology for efficiently planning the path for a fleet of shuttles in a tier-captive multi-deep four-way S/R system. Figure 3 illustrates the proposed methodological framework. The proposed framework was developed by decomposing the multi-agent routing problem into a sequence of decision-making stages reflecting the operational logic of a four-way shuttle system. First, transaction and layout information are collected and structured. Second, the operational parameters affecting vehicle interactions are defined. Third, an optimal route is generated for each vehicle through the A* algorithm. Finally, because multiple vehicles operate simultaneously in a shared environment, a dedicated collision-management layer is introduced. This layer is further divided into collision detection, priority assignment, and collision avoidance, allowing conflicts to be identified and resolved while preserving system throughput. The resulting framework combines route optimization and traffic management within a unified methodology suitable for multi-agent four-way shuttle operations.

The proposed methodology consists of four steps. The Data Entry step (Step 1 in Figure 3) contains the list of S/R transactions. Each transaction has the following features:

- Identification number.
- Item representing the UL (see the definition of "item" in Section 1).
- Entry time in the system.
- Type (S or R).
- Location, i.e., coordinates x, y, and z (see Figure 1).
- Lift for inbound (UL ingoing) and outbound (UL outgoing).

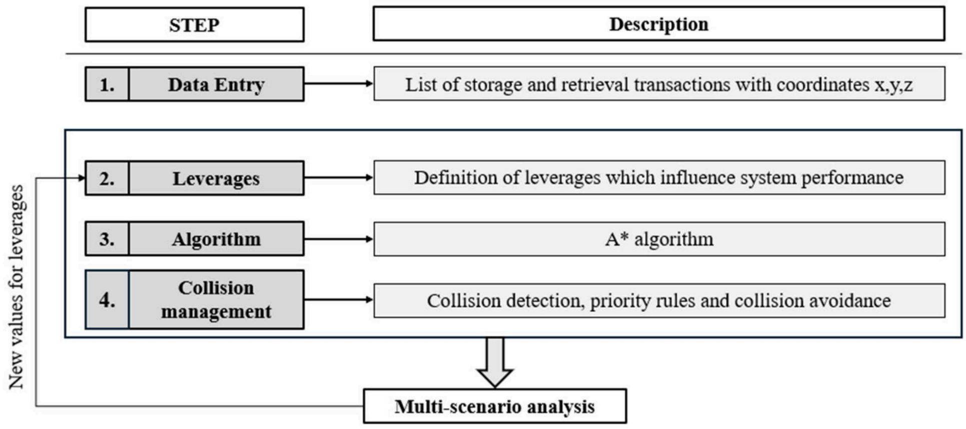


**Figure 3.** Methodology framework for Path Planning.

In a storage transaction, the path traveled by the vehicle comprises two travels. During the first travel, the vehicle starts from its current location and moves toward the inbound Lift, which transports the UL. During the second travel, the vehicle moves from the Lift to the transaction coordinates (x, y, and z). The path of a retrieval transaction is the opposite; the first travel starts from the current location of the vehicle and ends at the transaction coordinates. In the second travel, the UL is transported to the outbound Lift.

Step 2 defines the leverages that affect system performance. In particular, the number of vehicles in each tier is expected to be one of the most critical leverages because it influences both system productivity and the complexity of traffic management owing to congestion and collision avoidance. The other leverages are discussed below.

Step 3 presents the A* algorithm, which is used for path planning. The choice of the A* algorithm is motivated not only by its well-established capability to generate optimal paths, but also by its suitability for the specific operational characteristics of four-way shuttle-based storage and retrieval systems. In the considered environment, vehicles operate on a structured grid network composed of cross-aisles, storage lanes, and auxiliary aisles, where routing decisions must be updated frequently due to the simultaneous presence of multiple shuttles and the possibility of dynamic traffic conflicts. Compared with exhaustive search methods, A* significantly reduces the computational burden by exploiting an admissible heuristic while still guaranteeing optimal solutions. This characteristic is particularly important in real-time warehouse-control applications, where routing decisions must be generated rapidly to avoid delays in storage and retrieval operations.

Furthermore, the proposed collision-management framework requires repeated path recalculations whenever conflicts are detected. In this context, A* provides an effective balance between solution quality and computational efficiency, making it suitable for iterative re-planning procedures. Unlike more sophisticated optimization or learning-based approaches, which may require extensive training data, high computational effort, or complex parameter tuning, A* can be directly integrated into warehouse-control systems while maintaining transparency, robustness, and predictable computational behavior. These characteristics make A* particularly appropriate for the multi-agent routing problem investigated in this study.

The flowchart in Figure 4 illustrates the proposed multiple-vehicle routing methodology for a multi-tier system using path planning and collision management. This methodology works with discrete time steps. In Figure 4, t is the generic time unit/step in which the A* algorithm finds the best paths using collision management strategies in case of congestion.

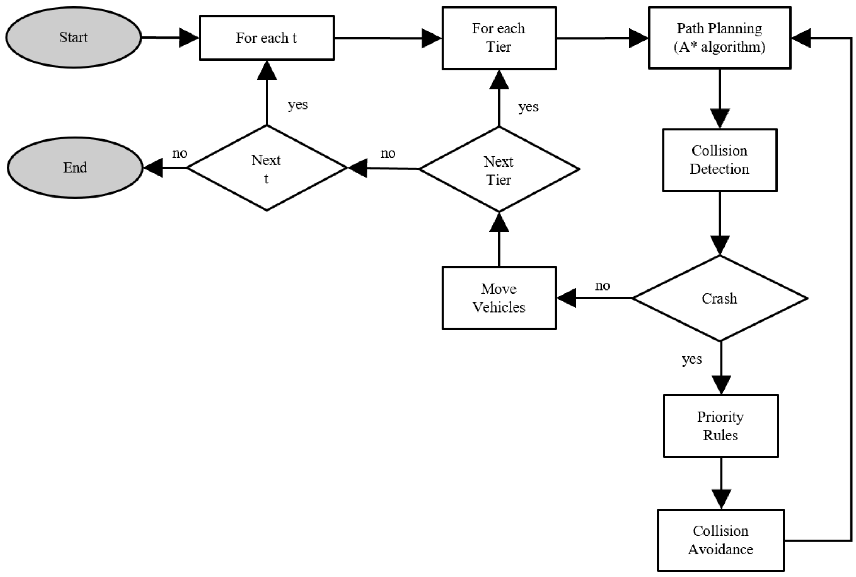


**Figure 4.** Flowchart of path planning and collision management.

A detailed description of the flow-chart is following:

- Start. The methodology begins.
- For each time step (t). The algorithm iterates through discrete time steps.
- For each tier. At each time step, the algorithm processes every tier in the system.
- Path planning (A* algorithm). For each tier, the paths for vehicles are planned using the A* algorithm, which finds the optimal path between a start and an endpoint. This nominal path is evaluated when a S/R transaction is assigned to a vehicle.
- Collision Detection- The system checks for potential collisions between operating vehicles along the planned paths.
- Crash check. The vehicles move along their planned paths if no expected crashes are detected. If a crash is detected, a set of priority rules is applied.
- Priority Rules. When a collision is detected, priority rules determine which vehicle should proceed or adjust its nominal path.
- Collision Avoidance. Based on the priority rules, the system adjusts paths to prevent collisions, and the process returns to path planning.
- Move Vehicles. After collision management, vehicles are moved according to their nominal or modified paths.
- Next Tier check. The process checks if all tiers have been processed. If yes, it moves to the next time step. If no, it continues to the next tier within the current time step.
- Next Time Step check. The process verifies if more steps need to be processed in more time. If yes, it loops back to the next time step. If not, the process ends.
- End. The algorithm terminates once all time steps and tiers have been processed.

This flowchart integrates path planning, collision detection, and collision avoidance to ensure efficient and safe movement of vehicles in a multi-tier system. The following subsection describes the formulation of the A* algorithm within the selected SBS/RS context (Step 3 in Figure 3), including how the cost and heuristic functions are adapted to the unique characteristics of four-way vehicle movements. Section 3.2 explains the collision-management process (Step 4), including its crucial phases.

### *3.1. A* in FSS/RS*

The A* algorithm is a widely used graph-traversal and path-planning method that guarantees the discovery of the optimal path between start and goal nodes. It combines the strengths of Dijkstra's algorithm and the greedy best-first search by employing a heuristic function to efficiently guide the search. Specifically, A* uses a cost function $f(n)$ for each node $n$, defined as:

$$f(n) = g(n) + h(n), \quad (1)$$

where $g(n)$ represents the exact cost of the path from the start node to $n$ and $h(n)$ is the heuristic estimate of the cost from $n$ to the goal node.

Equation (1) is applied at each node in the search space to compute its total estimated cost, balancing the incurred travel cost ($g$) with the expected remaining cost ($h$). In this way, A* always expands the node with the lowest estimated total cost, combining optimality with search efficiency. Importantly, the heuristic function h is designed to never overestimate the actual cost of the path, ensuring that the A* algorithm always finds an optimal solution. This study adopts the Manhattan distance because the shuttle moves along grid-like aisles, and the sum of horizontal and vertical distances provides an accurate and efficient estimate.

The algorithm operates by maintaining two sets of nodes: open and closed lists. The open list contains nodes to be evaluated, prioritized by their $f(n)$ values, while the closed list stores nodes already processed. At each iteration, the algorithm selects the node with the lowest $f(n)$ value from the open list, expands it by generating its neighbors, and updates the lists accordingly. The process continues until the goal node is reached or the open list is exhausted.

In robotics, A* plays a crucial role in the path planning of autonomous agents operating in dynamic and constrained environments. Similarly, in the context of four-way SBS/RSs, the ability to compute optimal paths efficiently aligns well with the need to minimize travel time and energy consumption. By leveraging the heuristic to account for system-specific constraints such as traffic congestion or temporary obstructions, A* can dynamically adjust shuttle routes to maintain high operational throughput.

This algorithm was applied to a four-way shuttle system with a grid-like storage structure, as illustrated in Figure 2. The shuttle located at the starting position S must retrieve an item stored at position E. The warehouse is represented as a weighted graph, where each node corresponds to a potential shuttle location, and the edges represent the movement between adjacent locations, with weights proportional to the travel time. The shuttle calculates the optimal path from S to E using the A* algorithm. The cost function (1) is applied at each step, where g(n) is the actual travel cost from S to the current node n, and h(n) is the heuristic function that estimates the remaining cost to E.

The grid layout justifies the assumption of the Manhattan distance as it calculates the sum of the horizontal and vertical distances between nodes. As the algorithm progresses, it evaluates possible paths, prioritizing nodes with the lowest $f(n)$ values. Once E is reached, the algorithm outputs the shortest path, ensuring minimal travel time while accounting for real-time constraints. This adaptive approach enables efficient and reliable operations in dynamic environments, demonstrating A* suitability for FSS/RS.

The next subsection describes the last step (Step 4 in Figure 3) of the proposed methodology, focusing on the collision management strategies once the A* algorithm finds the best path.

### *3.2. Collision Management*

Efficient collision management is critical for operating a four-way shuttle system in which multiple shuttles operate simultaneously within shared aisles and cross-aisle networks, creating a high potential for collisions. The effective management of such scenarios is essential to ensure system safety, maintain operational throughput, and minimize delays caused by conflicts in shuttle movements.

To address these challenges, this study adopts a structured approach to collision management. This approach is based on three key phases: collision detection, priority assignment, and collision avoidance.

- Collision Detection. This involves identifying potential collision points between shuttles. This step requires real-time monitoring of the shuttle position and predicted trajectories. By leveraging the path-planning capabilities of the A* algorithm, the system can preemptively detect intersections where shuttles may converge.
- Priority Assignment. Once potential collisions are identified, priority rules are applied to determine which shuttle should proceed and which should yield. These rules are designed to minimize disruptions, while considering factors such as task urgency, shuttle proximity to the endpoint, and overall system efficiency.
- Collision Avoidance. This phase adjusts the paths of the shuttles involved to resolve conflicts by replanning routes using the A* algorithm. The adopted strategy does not foresee a velocity reduction or temporary stopping of a shuttle to allow others to pass.

By systematically addressing collisions by these three phases, the system ensures smooth and safe operations, even in high-density traffic scenarios. The following subsections will provide a detailed discussion of each phase.

#### 3.2.1. Collision Detection

This phase is crucial for identifying potential conflicts between shuttles. This study introduces a Safety Area (SA) to facilitate the detection process. The SA is the set of P future positions that a shuttle occupies over a defined time horizon, as determined by its planned path using the A* algorithm. This includes both spatial and temporal dimensions and encapsulates the shuttle-movement trajectory within the warehouse environment. The definition of the SA depends on the parameter P, which is the leverage that affects the system performance.

To detect collisions, the SAs of all shuttles are continuously monitored and analyzed. A collision is predicted if an intersection exists between the SAs of two or more shuttles in space and time. This intersection indicates that shuttles occupy the same physical location simultaneously, creating a conflict that requires resolution.

This proactive approach allows the system to preemptively identify potential collisions, enabling the subsequent phases of priority assignment and collision avoidance to act before conflict materializes. By integrating the Safety Areas with the path-planning outputs of the A* algorithm, the collision-detection process ensures a reliable foundation for maintaining safe and efficient shuttle operations.

Figure 5 shows an example of the application of collision detection to a set of two vehicles. The grid illustrated in Figure 5 depicts the dual-access lanes of the system, with open passages on both sides allowing shuttles to move freely in either direction. Initially, (1) vehicle 1 and vehicle 2 are in their start positions. In (2) and (3), each vehicle evaluates its best path using A* algorithm, represented by a colored arrow in Figure 5. The collision detection phase (4) finds two locations where the Safety Areas of vehicles (red and blue paths in Figure 5) cross. This means that one of the vehicles needs to re-evaluate its path following the next steps of collision management.

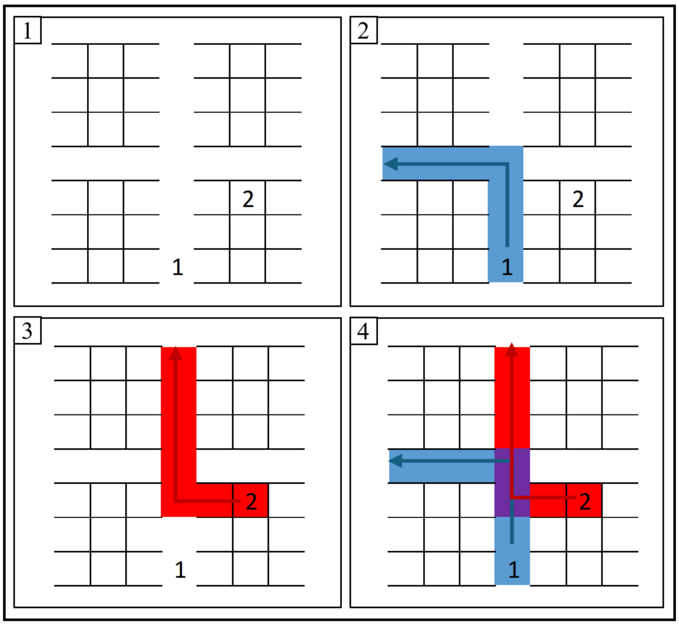


**Figure 5.** Example of collision detection application.

### 3.2.2. Priority Assignment

Once a potential collision is detected, the next phase consists of assigning priorities to the shuttles involved and determining which vehicle should proceed and which should yield. This was achieved by calculating a priority score for each shuttle by combining two key factors: shuttle-load status and distance to the destination. The first factor considers whether a shuttle is loaded. Loaded shuttles typically have higher operational importance because they actively complete storage or retrieval tasks. Thus, they have a higher priority for minimizing delays in fulfilling these critical operations. The second factor is the remaining number of nodes on the planned path of the shuttle to its destination. Shuttles closer to completing their journey are prioritized to avoid inefficient halts and disruptions in the overall system flow. These two contributions are combined into a single-priority score using the following weighted formula:

$$PS = \propto \times L + (1 - \propto) \times D, \quad (2)$$

where

- PS is the shuttle's priority score;
- L is the load factor with a value of 1 for loaded shuttles and 0 for unloaded shuttles;
- D represents the ranking of vehicles based on the number of steps required to reach their destinations. The vehicle with the most remaining steps is assigned D = 1, the second is assigned D = 2, etc. This ranking helps prioritize vehicles, with a greater D value indicating a higher priority (i.e., closer to reaching the destination);
- $\alpha$ is a weighting parameter ($0 \leq \alpha \leq 1$) that determines the relative importance of the two factors.

The $\alpha$ parameter allows the system to adapt priority assignments based on specific operational objectives. For example, increasing $\alpha$ places greater emphasis on the load status, favoring loaded shuttles, while reducing $\alpha$ prioritizes proximity to the destination.

By evaluating and comparing priority scores, the system ensures a fair and efficient resolution of conflicts, balancing throughput optimization with task urgency. This dynamic priority assignment mechanism seamlessly integrates with the subsequent collision avoidance phase to maintain smooth shuttle operations.

Following the example of Figure 5, it is assumed that both vehicles are loaded. Vehicle 2 is closer to its destination compared to vehicle 1. Considering $\alpha$ equal to 0.5, Equation (2) assigns a priority score PS equal to 1 to vehicle 1, while vehicle 2 has PS = 1.5. Since vehicle 2 has a higher priority, the path of vehicle one must be re-evaluated using the collision avoidance method.

### 3.2.3. Collision Avoidance

The final phase of the proposed collision management is collision avoidance, in which the system adjusts shuttle paths to resolve detected conflicts. After determining the priority of each vehicle involved in a potential collision, shuttles with lower priority scores must modify their paths. This is achieved by recalculating their routes using the A* algorithm, ensuring that the adjusted paths avoid collision points in both space and time.

To incorporate the collision information into the path re-planning process, the heuristic function $h$ in A* is dynamically modified. Specifically, the heuristic value is increased for the node and corresponding timestamp at which the collision is detected. By assigning a higher cost to the collision point in the spatiotemporal domain, the algorithm discourages the lower-priority vehicle from selecting that path, thereby effectively steering it toward an alternative collision-free route.

Figure 6 shows the evolution of Figure 5, incorporating the result of the collision avoidance method. The priority rule established that vehicle 1 has a lower priority and needs to re-evaluate its path, increasing the value of the heuristic function h in the violet nodes (4) in the unit time vehicle one should have been. In (5), vehicle 1 finds another collision-free path, avoiding the violet nodes, and the vehicles can proceed to their paths.

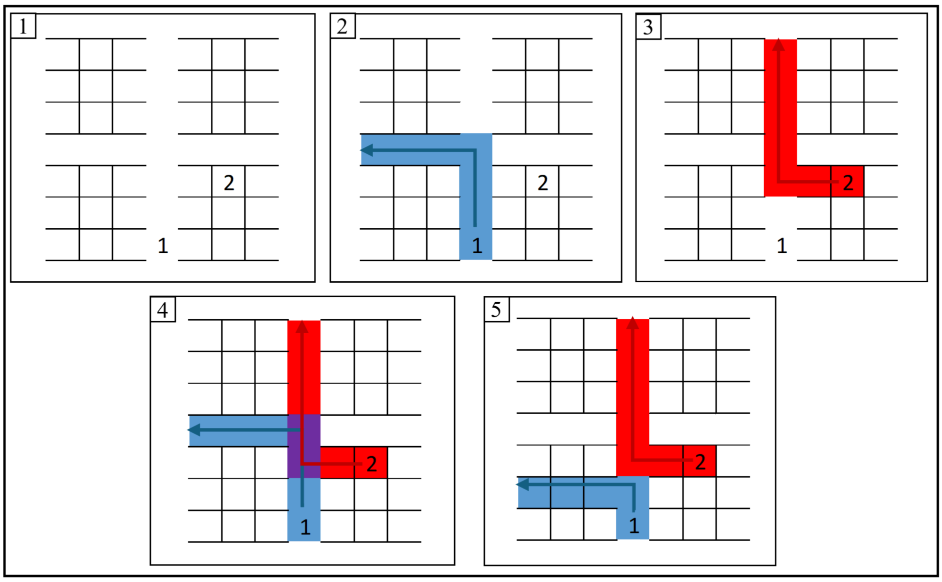


**Figure 6.** Example of collision management.

This adjustment ensures that the recalculated path accounts for both the warehouse's physical constraints and the shuttle movements' temporal dynamics. The A* algorithm efficiently identifies a new optimal path that minimizes travel time while avoiding the conflict area, thereby maintaining system safety and operational flow.

By leveraging the dynamic nature of the A* algorithm, the collision-avoidance phase ensures that conflicts are resolved in real time while maintaining operational efficiency. This proactive approach allows the system to adapt to changing conditions and maintain a high throughput, even in densely populated operational environments.

Table 2 lists the leverages identified in this study for the multi-scenario analysis conducted in Section 4.

**Table 2.** Leverages of multi-scenario analysis.

| Leverage | Description |
|---|---|
| Number of vehicles | The total number of shuttles operating simultaneously within a tier. |
| $\alpha$ | The weighting parameter is used in the priority score calculation, balancing the influence of load status and distance to the destination point. |
| $P$ | The size of the Safety Area represents the number of future positions considered for collision detection. |
| Entry time | This is the time at which each transaction enters the system. This study adopts a first data entry with an entry time equal to 0 and a second one with an entry time following an inter-arrival rate generated by an exponential distribution. |
| Collision detection | This leverage considers whether collisions are actively detected and managed. By controlling this parameter, a comparison can be made between the performances of a virtual system, in which shuttles move along their optimal paths as if collisions do not exist and vehicles pass through each other, and a real system, in which collisions are detected and resolved, forcing shuttles with lower priority to deviate from their optimal paths to avoid conflicts. |

These leverages allow for the flexible exploration of different operational scenarios, providing insights into how system-performance and collision-management strategies respond to different environments and operating conditions. Section 4 provides the results of a multi-scenario analysis of a numerical example that reproduces a real-world FSS/RS. Each scenario represents a unique configuration of system parameters. By simulating these scenarios, the methodology comprehensively assesses how each leverage impacts key performance outputs such as service and waiting times, throughput, vehicle idle time, etc.

## 4. Results

This section presents and discusses the results of a multi-scenario analysis generated by applying the proposed methodology to the data entry of 100 S/R transactions in a single tier of a multi-deep dual-access FSS/RS. Figure 7 shows a view of the tiered layout inspired by real applications in the food and beverage industry.

The tier comprises one central vertical aisle ($x$-axis in Figure 7) with 90 lanes on both sides. Additionally, two other vertical aisles ($x$-axis) are located on the left and right sides of the tier. Each side has nine blocks of lanes separated by auxiliary drive-through aisles ($z$-axis in Figure 7). Each lane has a depth of 11 locations, generating a total storage capacity of 1980 locations per tier. Six parking areas for shuttles, three per side, and four lifts at the corner of the tier, two for incoming ULs and two for outgoing ULs, are included. The shuttle is assumed to move with a constant velocity of 1 m/s. The portfolio of products counts six different items, represented by different colors (see the colored storage locations in Figure 7).

The data entry used in this multi-scenario analysis consists of an initial stock and a sequence of 100 S/R transactions. The initial storage assignment is generated randomly, in agreement with the previously introduced item homogeneity and dual-access assumptions.

Specifically, given a generic lane, a random cut is generated to define two semi-lanes. A random initial product storage is generated for each semi-lane based on its capacity.

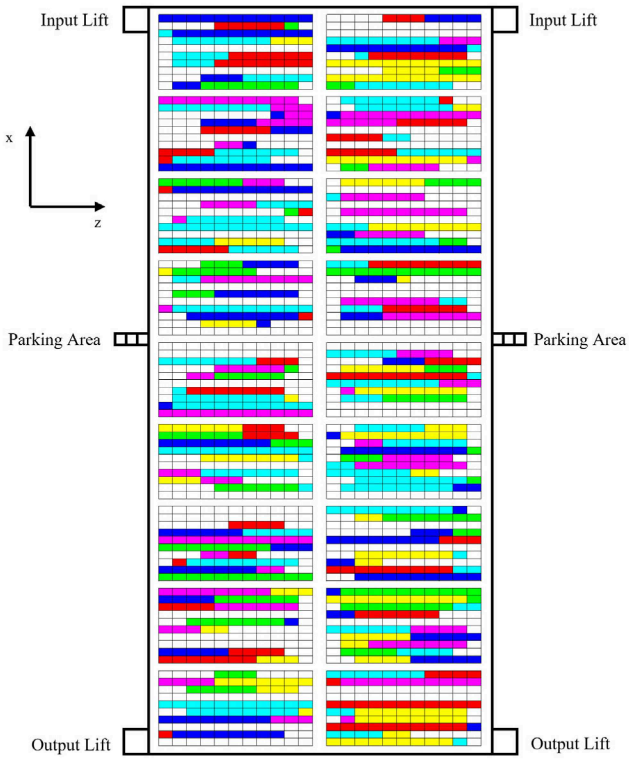


**Figure 7.** Tier layout of the numerical example.

Figure 8 shows an example of a block made of ten lanes with the following initial stock:

- Lane L1 has a virtual cut equal to 11 and is filled with the blue item;
- Lane L3 has a cut equal to 1. The left semi-lane is filled with the yellow item, and the right semi-lane contains 3 ULs of the blue item;
- Lane L5 has a virtual cut equal to 0 and is empty;
- Lane L6 has a cut equal to 5. The left semi-lane contains 4 ULs of the cyan item, and the right semi-lane contains 2 ULs of the blue item;
- Lane L7 has a cut equal to 4. The left is filled with the violet item, and the right semi-lane contains 6 ULs of the red item;
- Lane L11 has a cut equal to 3. The left semi-lane contains 1 UL of the red item, and the right semi-lane contains 7 ULs of the violet item.

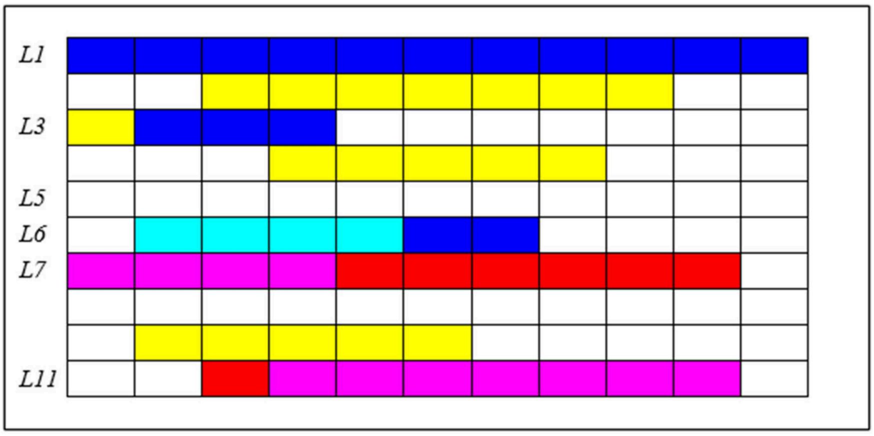


**Figure 8.** Example of initial storage generation.

This random assignment generates an initial stock of 1031 ULs distributed over the tier (see Figure 6), corresponding to a filling degree of 52%. Starting from this inventory level, the data entry counts 100 S/R transactions. Table 3 shows the features of the generic S/R transaction.

**Table 3.** Example of data entry transactions.

| ID | Time | Type | Item | X | Y | Z | Lift |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | Red | 2 | 34 | 1 | 1 |
| 2 | 17 | −1 | Blue | 15 | 13 | 1 | 3 |
| 3 | 31 | 1 | Green | 7 | 73 | 1 | 2 |
| 4 | 55 | −1 | Yellow | 21 | 47 | 1 | 4 |

The "ID" is the unique identifier of a transaction. The "Time" represents the entry time in the system. The "Type" equals 1 for a storage transaction and −1 for a retrieval transaction. The "Item" is represented by a color, as shown in Figure 2. The "XYZ" columns represent the location coordinates within the system. The "Lift" is the input/output lift identifier to enter/exit the system.

The assignment of the location to a storage transaction is a busy semi-lane hosting the same item in agreement with the item homogeneity constraint. If no lanes are available, the transaction searches for an empty lane. The location for a retrieval transaction belongs to a lane (or semi-lane) hosting the transaction item.

Once the initial stock and data entry are generated, a multi-scenario analysis of the system performance is conducted based on the proposed methodology.

Table 4 presents the leverage values assumed in the multi-scenario analysis.

**Table 4.** Leverage values for multi-scenario analysis.

| Leverage | Values | | | | | |
|---|---|---|---|---|---|---|
| Number of vehicles | 1 | 2 | 3 | 4 | 5 | 6 |
| $\alpha$ | 0 | 0.2 | 0.4 | 0.6 | 0.8 | 1 |
| $P$ | 5 | | 10 | | Max | |
| Entry time | Simultaneous | | | Exponential | | |
| Collision detection | Yes | | | No | | |

The number of vehicles ranged from 1 to 6. Leverage $\alpha$ ranged from 0 to 1 in steps of 0.2. The value "Max" for the priority score $P$ indicates that the SA equals the total path. The entry time can be equal to zero; thus, all sets of S/R transactions are ready to be processed simultaneously at the entry queue of the system (i.e., simultaneous scenario), or they can follow an inter-arrival time following an exponential distribution (i.e., exponential scenario). The exponential distribution has an average arrival rate of 120 transactions per hour. This was inspired by real-world applications. The collision-detection leverage may or may not be allowed, as shown in Table 2. A generic scenario is a simulation run that corresponds to a set of values, one for each leverage. A combination of these leverage levels generates 432 scenarios. The simulations were performed using MATLAB R2023b on a system equipped with an Intel Core i7-1165G7 processor and 8 GB RAM running Windows 11.

The performance output of the multi-scenario analysis quantified for each scenario include the simulation time, which is the makespan of the simulation run, average service time per transaction, average waiting time per transaction, total time to process the transactions, number of transactions processed per hour, and number of transactions processed per hour per vehicle.

Owing to the large number of scenarios, the discussion of the results is organized into five analyses, as illustrated in the following subsections. Each subsection focuses on a specific leverage. In particular:

- Analysis 1 shows the impact of the leverage $P$ used to evaluate the dimension of the Safety Area in collision detection;
- Analysis 2 shows the effect of the leverage $\alpha$ used to assess the priority of vehicles in case of collision;
- Analysis 3 shows the impact of the number of vehicles used to perform the S/R transactions;
- Analysis 4 shows the effect of varying the entry time of the S/R transactions;
- Analysis 5 shows the impact of the collision detection.

### *4.1. Analysis 1—Safety Area*

This analysis shows the result of applying the proposed methodology to a fleet of three shuttles and setting the priority-assignment parameter $\alpha$ to 0.6. The entry time of the S/R transactions follows an exponential distribution. Finally, the collisions are appropriately detected and solved. The assumed values of $P$, representing the number of future positions included in the SA evaluation (see Section 3.2.1), are 5, 10, and Max. The Max value of $P$ indicates that a generic vehicle on a route from the actual position to the destination position searches for one possible collision in the remaining future positions to the destination node.

Table 5 presents the results of Analysis 1. The impact of leverage $P$, an extension of the Safety Area, on system performance is as follows:

- The total simulation time, that is, the makespan of the system, decreases slightly as $P$ increases, but the difference is minimal (from 3702 s when $P$ is 5 to 3671 s when $P$ is Max). This suggests that increasing $P$ has a limited impact on the overall duration of the simulation.
- The average service time per transaction decreases as $P$ increases, going from 101.57 s when $P$ is 5 to 100.31 s when $P$ is Max. This indicates a slight improvement in efficiency owing to the more effective dispatching of transactions to the vehicles.
- The transaction waiting time follows a decreasing trend, from 267.03 s when $P$ is 5 to 260.55 s when $P$ is equal to Max. This suggests that increasing $P$ reduces the congestion and vehicle idle time, thereby leading to smoother operation.
- The throughput of the system improves slightly when $P$ increases from 97.24 to 98.07 transactions per hour. This implies that the system can handle a marginally higher number of tasks as $P$ increases.
- The number of collisions detected assumes similar values in the range 33 and 36.

**Table 5.** Results of Analysis 1—Safety Area.

| ***P*** | **Simulation Time [s]** | **AVG Service Time [s/Transaction]** | **AVG Waiting Time [s/Transaction]** | **Transactions per Hour** | **Collisions Detected** |
|---|---|---|---|---|---|
| 5 | 3702 | 101.57 | 267.03 | 97.24 | 35 |
| 10 | 3672 | 100.52 | 262.18 | 98.04 | 33 |
| Max | 3671 | 100.31 | 260.55 | 98.07 | 36 |

An increasing $P$ value results in a slight reduction in the total simulation time and average service and waiting times per transaction, indicating improved efficiency in transaction dispatching. The system throughput increases marginally, suggesting a slightly higher transaction-handling capacity. The number of detected collisions remains relatively stable, confirming that extending the Safety Area does not introduce significant operational safety concerns.

### *4.2. Analysis 2—Priority Assignment*

This analysis shows the results of applying the proposed methodology to a fleet of three shuttles using a $P$ value of 10. The entry times of the transactions follow an exponential distribution. The collisions are detected and solved. The leverage $\alpha$ can vary from 0 to 1 in steps of 0.2. The value of $\alpha$ represents the weight of the two contributions used to evaluate the vehicle priority in the case of a collision (see Section 3.2.2).

Table 6 presents the results of Analysis 2. The impact of leverage $\alpha$ on system performance is as follows:

- The simulation time stays relatively stable across all values of $\alpha$, with only a slight increase when $\alpha$ is 0.6 and 0.8. Increasing $\alpha$ gives more priority to the loaded vehicle; therefore, the system may adjust the flow slightly to prioritize loaded vehicles. However, the overall simulation time does not change dramatically.
- Initially, the average service time per transaction is consistent at 97.75 s for $\alpha$ equal to 0, 0.2, and 0.4. However, when $\alpha$ increases to 0.6 and 0.8, the service time increases to 100.52 s. When $\alpha$ is 1, the service time slightly improves to 98.90 s owing to the better optimization of the priority rules when the loaded vehicle is prioritized.
- The waiting time increases with higher $\alpha$ values, from 225.06 s at lower $\alpha$ values (0, 0.2, and 0.4) to 262.18 s when $\alpha$ is 0.6 and 0.8, before slightly decreasing to 248.68 s when $\alpha$ is 1. The lower waiting times at lower $\alpha$ values indicate that the system performs better when more priority is given to the loaded vehicle by encouraging the prioritization of loaded vehicles that completed the S/R transaction and are available to start the next one.
- The throughput decreases slightly as $\alpha$ increases. When $\alpha$ is 0 and 0.2, the system achieves 100.19 transactions per hour. At higher $\alpha$ values (0.6 and 0.8), throughput decreases to 98.04 transactions per hour. However, throughput increases slightly to 99.61 transactions per hour when $\alpha$ is 1.
- The number of collisions detected decreases as $\alpha$ increases. If $\alpha$ equals 0, 0.2, and 0.4, the number of collisions is 37. This number decreases to 33 when $\alpha$ is 0.6 and 0.8, and further to 30 at an $\alpha$ value of 1. This indicates that prioritizing loaded vehicles leads to fewer collisions.

**Table 6.** Results of Analysis 2—Priority Assignment.

| $\alpha$ | Simulation Time [s] | AVG Service Time [s/Transaction] | AVG Waiting Time [s/Transaction] | Transactions per Hour | Collisions Detected |
|---|---|---|---|---|---|
| 0 | 3593 | 97.75 | 225.06 | 100.19 | 37 |
| 0.2 | 3593 | 97.75 | 225.06 | 100.19 | 37 |
| 0.4 | 3593 | 97.75 | 225.06 | 100.19 | 37 |
| 0.6 | 3672 | 100.52 | 262.18 | 98.04 | 33 |
| 0.8 | 3672 | 100.52 | 262.18 | 98.04 | 33 |
| 1 | 3614 | 98.90 | 248.68 | 99.61 | 30 |

This analysis shows that increasing $\alpha$ impacts system performance by slightly increasing the simulation and service times while leading to higher waiting times when $\alpha$ is 0.6 and 0.8. Throughput decreases slightly as $\alpha$ increases, but improves at $\alpha$ equal to 1. Notably, collisions decrease with higher $\alpha$ values, indicating that prioritizing loaded vehicles enhances safety and reduces conflicts.

*4.3. Analysis 3—Number of Vehicles*

This analysis shows the results of applying the proposed methodology to a fleet with a varying number of shuttles. This analysis uses a leverage $\alpha$ equal to 0.6 for priority assignment and leverage $P$ equal to 10 for collision detection. The entry time of the transactions follows an exponential distribution, and collisions are detected and solved. In this analysis, the number of vehicles varies from one to six.

Table 7 presents the results of Analysis 3. The impact of the number of vehicles on the system performance is as follows:

- As the number of vehicles increases from one to six, the simulation time decreases. The system becomes more efficient with more vehicles, as it can process the entire set of transactions more quickly (from 9712 s when the number of vehicles is one to 3075 s when the number of vehicles is four). However, a slight difference exists between the results for four, five, and six vehicles, demonstrating that many vehicles may introduce overhead or dispatching complexity.
- The average service time per transaction decreases slightly from 96.23 s for one vehicle to 94.98 s for two vehicles. However, it increases for three vehicles (100.52 s) and then fluctuates around 96–97 s as more vehicles are added. This suggests that the number of vehicles slightly affects the average service time of transactions.
- The average waiting time decreases as the number of vehicles increases. In the case of one vehicle, the waiting time is the highest at 3200.15 s, likely due to bottlenecks caused by insufficient vehicles to meet the demand. As more vehicles become available, the waiting time decreases significantly, reaching just 0.87 s for six vehicles.
- As the number of vehicles increases, the number of transactions per hour rises significantly, from 37.07 transactions per hour with one vehicle to 117.07 transactions per hour with four vehicles. This reflects the increased capacity of the system to handle more tasks when more vehicles are available. After four vehicles, a slight difference in the transactions per hour is observed for five and six vehicles (117.11 and 116.32 transactions per hour, respectively), indicating that further increasing the number of vehicles may have a diminishing return regarding transaction throughput.
- The number of detected collisions increases with the number of vehicles, reaching 79 for six vehicles. This suggests that as the number of vehicles increases, the risk of collisions also increases, likely owing to more complex interactions and less-efficient traffic management.

**Table 7.** Results of Analysis 3—Number of vehicles.

| Vehicles | Simulation Time [s] | AVG Service Time [s/Transaction] | AVG Waiting Time [s/Transaction] | Transactions per Hour | Collisions Detected |
|---|---|---|---|---|---|
| 1 | 9712 | 96.23 | 3200.15 | 37.07 | / |
| 2 | 4927 | 94.98 | 849.23 | 73.07 | 15 |
| 3 | 3672 | 100.52 | 262.18 | 98.04 | 33 |
| 4 | 3075 | 95.06 | 27.18 | 117.07 | 46 |
| 5 | 3074 | 96.84 | 4.10 | 117.11 | 81 |
| 6 | 3095 | 97.32 | 0.87 | 116.32 | 79 |

Increasing the number of vehicles leads to improved system throughput and reduced transaction waiting time because more vehicles allow for a more efficient dispatching of S/R tasks. However, large fleets also result in an increased number of collisions, suggesting a trade-off between system throughput and safety. Four vehicles guarantee the minimum makespan, that is, simulation time. In particular, new vehicles do not further improve the system performance. Finally, a suitable vehicle count exists to balance the throughput and safety.

To further investigate the impact of transaction arrival patterns on system performance, an additional analysis extends the previous study by introducing multiple arrival scenarios for different workloads. In the original configuration, the entry times of the transactions were generated using a negative exponential distribution with an average rate (lambda) of 120 transactions per hour. Four additional random sequences were generated using the same statistical distribution, resulting in five alternative set of data entry that represent different runs of the same stochastic arrival process. The performance metrics obtained from these five simulations were then averaged to obtain more robust and less noise-sensitive estimates.

The same procedure was repeated for two additional arrival rates, namely 60 transactions per hour and 240 transactions per hour, again varying the number of vehicles from one to six. Table 8 presents the aggregated results of this extended analysis. By comparing different arrival intensities and multiple stochastic scenarios, this analysis provides a more comprehensive understanding of how system performance scales with workload variability and fleet size.

**Table 8.** Results of the extended analysis with multiple exponential-arrival scenarios and different average arrival rates.

| l | Vehicles | Simulation Time [s] | AVG Service Time [s/Transaction] | AVG Waiting Time [s/Transaction] | Transactions per Hour | Collisions Detected |
|---|---|---|---|---|---|---|
| 120 | 1 | 9690.4 | 96.04 | 3290.58 | 37.14 | / |
| | 2 | 4863.6 | 93.97 | 926.27 | 74.02 | 17.4 |
| | 3 | 3473 | 98.02 | 253.49 | 103.78 | 29.6 |
| | 4 | 3002 | 96.99 | 33.95 | 120.09 | 47.8 |
| | 5 | 2972.8 | 96.96 | 7.99 | 121.30 | 77.6 |
| | 6 | 2961.8 | 97.66 | 2.66 | 121.77 | 103.6 |
| 60 | 1 | 9747.6 | 96.02 | 1903.17 | 36.93 | / |
| | 2 | 5816.6 | 95.32 | 77.2 | 61.98 | 23 |
| | 3 | 5726.6 | 94.44 | 11.29 | 62.99 | 34.2 |
| | 4 | 5742 | 93.78 | 2.46 | 62.82 | 39 |
| | 5 | 5729.2 | 93.58 | 0.46 | 62.97 | 51.4 |
| | 6 | 5717.8 | 92.15 | 0.12 | 63.09 | 48.4 |
| 240 | 1 | 9682 | 96.15 | 4004.04 | 37.18 | / |
| | 2 | 4823 | 94.28 | 1604.61 | 74.64 | 28.8 |
| | 3 | 3398.8 | 99.35 | 892.98 | 105.93 | 28.6 |
| | 4 | 2630.6 | 99.58 | 512.38 | 136.86 | 43 |
| | 5 | 2157.8 | 98.53 | 273.06 | 166.87 | 69.4 |
| | 6 | 1946.4 | 101.91 | 148.05 | 185.05 | 84.2 |

The results of the extended analysis show how system performance varies with both the number of vehicles and the transaction arrival rate. Simulation time, or makespan, exhibits a strong dependence on fleet size, particularly under medium and high workload conditions. For the scenario with an average arrival rate of 120 transactions per hour, simulation time decreases sharply when increasing the fleet from one to two vehicles, dropping from 9690.4 s to 4863.6 s. Further reductions are observed up to four vehicles, after which the improvement becomes marginal, indicating that four vehicles are sufficient to process this workload efficiently and that adding more vehicles introduces minimal benefits, possibly due to coordination overhead. In the low-load scenario with 60 transactions per hour, the most significant improvement occurs between one and two vehicles, from

9747.6 s to 5816.6 s, whereas adding further vehicles up to six barely affects simulation time, highlighting that under low demand the system is quickly relieved of bottlenecks. Conversely, under the high-load scenario of 240 transactions per hour, simulation time continues to decrease substantially across the full range of fleet sizes, reaching 1946.4 s with six vehicles, showing that the system remains capacity-constrained even with a larger number of shuttles.

Average service time per transaction remains relatively stable across all scenarios, fluctuating between 92 and 101 s, which indicates that the execution of individual transactions is largely independent of fleet size. In contrast, average waiting time is highly sensitive to both the number of vehicles and the workload. Under medium workload, waiting time is extremely high with a single vehicle, exceeding 3200 s, and decreases rapidly with additional vehicles, becoming almost negligible at five and six vehicles. In the low-load scenario, waiting time is already moderate with one vehicle and approaches minimal levels with three or more vehicles. High-load conditions result in very long waiting times for small fleets, with values above 4000 s for one vehicle and still significant delays at four vehicles, highlighting the need for a large fleet to handle heavy traffic efficiently.

Transaction throughput mirrors these trends. For medium workload, throughput rises sharply with the number of vehicles and saturates around 121 transactions per hour for four or more vehicles, reflecting the arrival rate limit. Under low load, throughput plateaus around 63 transactions per hour beyond two vehicles, confirming that additional capacity does not increase processing beyond the incoming demand. Under high load, throughput continues to increase with fleet size, reaching 185 transactions per hour with six vehicles, showing that the system can still exploit additional vehicles to process more transactions. However, the number of detected collisions consistently grows with the number of vehicles in all scenarios, reflecting the increased complexity of vehicle interactions and highlighting the trade-off between efficiency and safety.

Overall, these results demonstrate that fleet size has a strong impact on both system efficiency and congestion. For low-load scenarios, a small number of vehicles is sufficient to achieve minimal waiting times and near-optimal throughput. Medium-load scenarios reach optimal performance with around four vehicles, while high-load conditions require five to six vehicles to maintain reasonable processing times and throughput. At the same time, larger fleets inherently generate more collisions, suggesting the existence of an optimal fleet size that balances operational efficiency with safe vehicle interactions.

A factorial analysis of variance (ANOVA) was performed to assess whether the experimental results are statistically significant. The arrival rate $\lambda$ and the number of vehicles (V) are fixed factors in the analysis for the performance indicators reported in Table 8 (i.e., simulation time, AVG service time, AVG waiting time, transactions per hour, and collisions detected).

Table 9 illustrates the adopted model structure composed of three different analyses:

- Analysis $\lambda$. The main effect of $\lambda$, whose levels are 60, 120, and 240 transactions/hour.
- Analysis V. Main effect of number of vehicles V, whose levels vary from 1 to 6.
- Analysis $\lambda$ x V. Interaction effect capturing whether the impact of fleet size changes depends on workload intensity.

Table 9 reports the F statistic and corresponding *p*-value for the three analyses, combined with the performance indicators, to illustrate the effects of the selected factors on operational time, congestion, throughput, and safety-related outcomes.

The ANOVA results reported in Table 9 provide strong statistical evidence that system performance is significantly affected by $\lambda$, V, and their interaction. However, the relative importance of these factors varies substantially among the performance indicators. Figure 9 reports the interaction effects plots derived from the ANOVA results.

**Table 9.** ANOVA analysis across performance indicators.

| | Simulation Time [s] | | AVG Service Time [s/Transaction] | | AVG Waiting Time [s/Transaction] | | Transactions per Hour | | Collisions Detected | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Analyses** | **F** | ***p*** | **F** | ***p*** | **F** | ***p*** | **F** | ***p*** | **F** | ***p*** |
| l | 1652 | <0.001 | 56.39 | <0.001 | 988 | <0.001 | 2528 | <0.001 | 12.15 | <0.001 |
| V | 3064 | <0.001 | 7.01 | <0.001 | 3238 | <0.001 | 1487 | <0.001 | 106.62 | <0.001 |
| L x V | 112 | <0.001 | 9.14 | <0.001 | 130 | <0.001 | 296 | <0.001 | 6.31 | <0.001 |

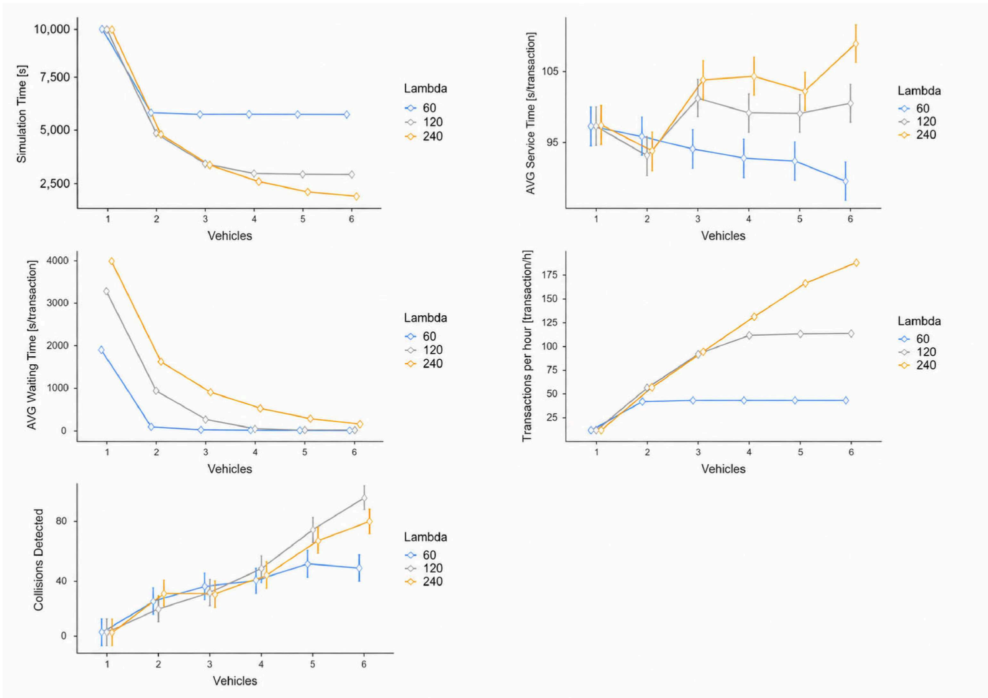


**Figure 9.** Interaction effects plots for simulation time, AVG service time, AVG waiting time, transactions per hour, and collisions detected.

The plots show that fleet size V emerges as the dominant factor for time- and congestion-related indicators, particularly simulation time and waiting time. Simulation time decreases sharply when moving from one to two vehicles and then quickly stabilizes at a low value of λ (i.e., 60). At the same time, it continues to decrease with increasing fleet size and medium/high arrival rates (i.e., 120 and 240). Average service time shows only minor variation across fleet sizes, confirming that it is largely determined by intrinsic transaction characteristics (e.g., transaction coordinates). In contrast, waiting time is highly sensitive to both factors, dropping rapidly as fleet size increases, especially under higher arrival rates. Throughput increases with fleet size but stabilizes early at low arrival rates, while continuing to grow at high workloads. Finally, collisions consistently increase with the number of vehicles, revealing the trade-off between efficiency gains and increased interaction complexity. Overall, the trends illustrated in Figure 9 are fully consistent and visually reinforce the quantitative findings reported in Tables 7 and 8.

### *4.4. Analysis 4—Entry Time*

This analysis presents the results of applying the proposed methodology to a fleet of shuttles by varying the S/R transaction-entry time. This analysis assumes an α (priority assignment) of 0.6 and P (SA extension) of 10. In this scenario, S/R transactions virtually appear simultaneously at the beginning of the simulation (i.e., "simultaneous scenario"). Collisions are detected and solved. In this analysis, the number of vehicles varies from one to six. This variation is conducted to maintain consistency with Table 7 and to allow a clear comparison of the impact of entry times on system performance. While the primary focus is on entry time effects, varying the fleet size helps illustrate how different numbers of vehicles influence the observed trends.

A comparison of Table 7 with Table 10 indicates important considerations regarding the impact of transaction entry time. The following points should be considered in particular:

- With a single vehicle, the simulation times are nearly identical in both cases (approximately 9700 s). As more vehicles are added, the system performs all transactions much faster in the simultaneous scenario (e.g., 2125 s for five vehicles versus 3074 s, assuming the real entry time). This suggests that when all transactions enter simultaneously, the system quickly processes the backlog, and the makespan is reduced. In contrast, in the exponential scenario, the system remains active for a longer time owing to the processing of new arrivals, incoming tasks, and outgoing (retrieval) tasks.
- The average waiting time per transaction is much higher in the simultaneous scenario, especially for lower vehicle counts (e.g., 4725.99 s vs. 3200.15 s for one vehicle, 2300.18 s vs. 849.23 s for two vehicles, and 1597.58 s vs. 262.18 s for three vehicles). Because all transactions are introduced simultaneously, many must wait a long time before a vehicle becomes available. In contrast, the exponential scenario introduces transactions gradually and increases waiting times. With more vehicles, waiting times decrease in both cases, but remain consistently higher in the simultaneous scenario owing to the initial congestion.
- The number of transactions per hour is significantly higher with simultaneous arrivals because the system can process more tasks in a shorter time frame. Six vehicles perform 193.86 transactions per hour in the simultaneous scenario, whereas in the exponential scenario, they perform 116.32 transactions per hour.
- More collisions occur in the simultaneous scenario involving a large fleet of vehicles (e.g., 98 collisions for six vehicles versus 79 in the exponential scenario). Immediate congestion at time 0 increases the probability of conflict. In contrast, the exponential scenario introduces transactions gradually, leading to a smoother traffic flow.

**Table 10.** Results of Analysis 4—Entry time.

| Vehicles | Simulation Time [s] | AVG Service Time [s/Transaction] | AVG Waiting Time [s/Transaction] | Transactions per Hour | Collisions Detected |
|---|---|---|---|---|---|
| 1 | 9682 | 96.24 | 4725.99 | 37.18 | / |
| 2 | 4778 | 93.67 | 2300.18 | 75.35 | 17 |
| 3 | 3362 | 99.18 | 1597.58 | 107.08 | 36 |
| 4 | 2595 | 99.87 | 1204.13 | 138.73 | 39 |
| 5 | 2125 | 98.97 | 945.45 | 169.41 | 61 |
| 6 | 1857 | 102.98 | 793.29 | 193.86 | 98 |

In the simultaneous scenario, the system achieves a significantly higher level of productivity because all transactions are available from the start, allowing the vehicles to operate at full capacity without idle periods. However, this generates an initial critical congestion, which leads to longer waiting times and greater collisions. In contrast, the exponential scenario distributes the workload more gradually, reducing waiting times and collisions, but resulting in a lower overall throughput. In the simultaneous scenario, the productivity steadily increases to six vehicles without showing clear signs of saturation. This suggests that the system may still benefit from additional vehicles, unlike in the exponential scenario, where performance gains start to level off at approximately four vehicles. The absence of a sharp decline in efficiency indicates that further strain on the system by the addition of more vehicles may potentially lead to an even higher throughput. However, this would likely result in increased congestion and collision rates, requiring a careful balance between maximizing productivity and maintaining operational stability.

### *4.5. Analysis 5—Collision Detection*

This analysis shows the results of applying the proposed methodology to a fleet of shuttles without considering the impact of collisions. This analysis assumes an $\alpha$ (priority assignment) of 0.6 and P (SA extension) of 10. The entry times of the transactions follow an exponential distribution. In this analysis, the collision-detection phase is disabled. In this manner, each vehicle follows the best path according to the A* algorithm without requiring path re-evaluation in the case of collision. In this analysis, the number of vehicles varies from one to six. Table 11 presents the results of Analysis 5.

**Table 11.** Results of Analysis 5—Focus on collision detection.

| Vehicles | Simulation Time [s] | AVG Service Time [s/Transaction] | AVG Waiting Time [s/Transaction] | Transactions per Hour | Collisions Detected |
|---|---|---|---|---|---|
| 1 | 9712 | 96.23 | 3200.15 | 37.07 | / |
| 2 | 4861 | 93.65 | 845.41 | 74.06 | / |
| 3 | 3525 | 65.62 | 220.62 | 102.13 | / |
| 4 | 3072 | 93.54 | 20.42 | 117.19 | / |
| 5 | 3075 | 94.17 | 4.18 | 117.07 | / |
| 6 | 3074 | 90.90 | 0.35 | 117.11 | / |

A comparison of Table 11 with Table 7 shows that where collisions are detected and managed, the results are remarkably similar. The productivity of transactions per hour remains nearly identical for each vehicle count, and the service and waiting times show only minimal differences. This demonstrates that the proposed collision-management methodology introduces negligible overhead, allowing the system to closely approximate the virtual optimum, where each vehicle follows its theoretical path without interference.

In particular, even with collision handling, the system achieves nearly the same throughput, indicating that the proposed multi-agent methodology is effective in real applications, minimizing disruptions while ensuring safe and coordinated vehicle movements. This reinforces the robustness of the approach, proving that the system can efficiently balance optimal path planning and real-time traffic management without significant losses in system performance, which is traditionally measured as throughput.

## 5. Discussion

This is the first study to investigate the multiagent path-planning problem within deep-lane and dual-access four-way SBS/RSs. The proposed methodology and the multi-scenario analysis conducted on an application inspired by a real case study provide significant managerial insights for the design and operation of advanced automated storage systems.

The study supports strategic decision-making aimed at improving system performance and provides a detailed understanding of the tradeoffs between efficiency, congestion management, and operational robustness. These represent crucial aspects for warehouse designers and managers involved in complex automated storage environments.

The contribution of this study differs from most existing works on four-way shuttle systems. Previous studies have primarily addressed task scheduling, resource allocation, and operational optimization. For example, Refs. [29,31,36,41] focused on scheduling and task-allocation problems using metaheuristic or reinforcement-learning approaches. Similarly, Refs. [37,38] investigated scheduling optimization in four-way shuttle systems with multiple lifts, while Ref. [39] proposed a graph-based cooperative path-planning framework. Although these studies provide important contributions, they investigate different system architectures, operational assumptions, and optimization objectives. In particular, none of them specifically addresses the interaction between multi-agent path planning and a heuristic collision-management methodology in a four-way shuttle system characterized by multi-deep dual-access lanes and homogeneous storage policies. Consequently, a direct quantitative benchmark would not provide a meaningful comparison. The contribution of the present work therefore lies in the development and analysis of a dedicated methodology for a system configuration that remains largely unexplored in the current literature.

Four-way systems are distinguished by their flexibility and adaptability, particularly in high-density storage contexts. The capability of multidirectional movement reduces S/R cycle times and improves space utilization. Moreover, the implementation of dual-access multi-deep lanes further increases operational flexibility by enabling the storage of two different types of ULs within the same lane and allowing access from both ends. This configuration improves storage density without compromising operational performance, making FSS/RSs particularly suitable for modern supply-chain requirements.

The role of intelligent control methodologies and integrated WMS solutions in automated storage and material-handling systems is becoming increasingly important. In this context, the proposed multiagent path-planning methodology based on the A* algorithm allows the identification of efficient vehicle routes while minimizing delays and improving overall system throughput. The methodology combines computational efficiency and cost-effectiveness by evaluating alternative paths and selecting those characterized by the lowest travel costs. Furthermore, the proposed multi-agent approach effectively manages potential obstacles and vehicle collisions.

The results demonstrate that the number of vehicles strongly affects both productivity and traffic management. Although increasing the shuttle fleet generally improves throughput, it also introduces higher traffic complexity, congestion, and collision risks. Therefore, collision management emerges as a critical issue requiring structured strategies based on collision detection, priority assignment, and collision avoidance mechanisms. Real-time monitoring of vehicle positions and trajectories, combined with effective priority rules, represents an essential component for ensuring smooth traffic flow and operational stability.

The multi-scenario analysis, consisting of 432 scenarios, highlights the impact of several operational parameters on system performance, including fleet size, the weighting of priority factors ($\alpha$), Safety Area size (P), entry time management, and collision-detection policies. The obtained results provide practical insights for improving the design and management of FSS/RSs and contribute to a deeper understanding of the interactions among operational levers and their combined effects on system behavior.

Despite the contributions of this study, some limitations should be acknowledged. First, the numerical analysis was conducted on a single-tier four-way shuttle system inspired by a realistic industrial application. Although the considered layout is representative of real-world environments, different warehouse configurations may lead to different operational behaviors. Second, the analysis focused on fleet sizes up to six vehicles per tier, and therefore the computational scalability of the proposed methodology in environments with dozens or hundreds of simultaneously operating shuttles was not investigated. Third, the vehicle dynamics were simplified by assuming constant shuttle speed and neglecting acceleration and deceleration phases. Finally, the study focused on an A*-based routing framework and did not include direct comparisons with alternative path-planning approaches. These limitations provide opportunities for future research and further methodological developments.

## 6. Conclusions

This study proposed a multiagent path-planning methodology for deep-lane dual-access four-way SBS/RSs and demonstrated its applicability through a comprehensive multi-scenario analysis inspired by a realistic industrial case study. The proposed approach contributes to bridging the gap between academic research and industrial practice by considering realistic operational assumptions and addressing practical traffic-management challenges in four-way shuttle systems.

The findings confirm the importance of balancing throughput improvement and congestion control when increasing the shuttle fleet size. The study also demonstrates the relevance of structured collision-management strategies and intelligent routing methodologies in ensuring efficient and robust system operations.

Several future research directions emerge from this work. First, future studies may investigate optimization algorithms for determining the optimal length of multi-deep homogeneous lanes and semi-lanes, considering the impact of assignment strategies on storage and retrieval efficiency. Another possible development concerns the comparison of FSS/RSs with alternative automated storage technologies such as AVS/RSs to support technology-selection decisions.

Further research may also focus on tier-to-tier shuttle systems capable of both horizontal and vertical movements, as well as four-way tier-to-tier configurations integrating lift systems and advanced routing strategies. Additionally, introducing acceleration and deceleration phases into the vehicle dynamics could improve the realism and accuracy of cycle-time estimations.

Future developments could also involve advanced collision-management models considering variable shuttle speeds, inter-vehicle communication, and predictive movement strategies to improve traffic coordination and operational smoothness.

An additional research direction concerns the evaluation of the computational scalability of the proposed methodology in large-scale warehouse environments characterized by dozens or hundreds of simultaneously operating shuttles. Future studies could investigate computational performance, solution times, and scalability limits under increasingly complex traffic conditions.

Finally, the integration of artificial intelligence and machine-learning techniques represents a promising research direction for enhancing path planning, traffic management, demand forecasting, and adaptive real-time control strategies in next-generation automated warehouse systems.

**Author Contributions:** Conceptualization, G.L., A.L. and R.M.; methodology, G.L., A.L. and R.M.; validation, G.L., R.M. and G.S.; formal analysis, G.L. and G.S.; investigation, G.L.; data curation, G.L. and G.S.; writing—original draft preparation, G.L.; writing—review and editing, G.L., R.M. and G.S.; visualization, G.L. and G.S.; supervision, A.L. and R.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Acknowledgments:** During the preparation of this manuscript, the authors used Grammarly and ChatGPT (OpenAI v5.5) for the purpose of improving the English language, grammar, and readability of the text. The authors reviewed and edited all suggestions generated by these tools and take full responsibility for the content of this publication.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| FSS/RS | Four-way shuttle-based storage and retrieval system |
| SBS/RS | Shuttle-based storage and retrieval system |
| S/R | Storage or retrieval |
| AVS/RS | Automated vehicle storage and retrieval system |
| UL | Unit Load |
| LIFO | Last In First Out |
| WMS | Warehouse management system |
| AS | Assignment strategy |
| LD | Lane Depth |
| SL | Storage location |
| ST | System throughput |
| AS/RS | Automated storage and retrieval system |
| AMR | Autonomous mobile robot |
| RL | Reinforcement learning |
| AGV | Automated guided vehicle |
| PPO | Proximal Policy Optimization |
| IDQN | Independent deep queueing network |
| WT | Waiting time |
| IT | Idle time |
| CD | Collision detection |
| SA | Safety Area |
| AI | Artificial intelligence |
| ML | Machine learning |

## References

1. Azadeh, K.; Koster, R.; Roy, D. Robotized and automated warehouse systems: Review and recent developments. *Transp. Sci.* **2019**, *53*, 917–945. [CrossRef]
2. Ekren, B.Y.; Heragu, S.S. Simulation-based performance analysis of an autonomous vehicle storage and retrieval system. *Simul. Model. Pract. Theory* **2011**, *19*, 1640–1650. [CrossRef]
3. Battarra, I.; Accorsi, R.; Manzini, R.; Rubini, S. A hybrid model for the design of a deep-lane multi satellite AVS/RS. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 1191–1217. [CrossRef]
4. Lupi, G.; Accorsi, R.; Battarra, I.; Manzini, R.; Sirri, G. Space efficiency and throughput performance in AVS/RS under variant lane depths. *Int. J. Adv. Manuf. Technol.* **2024**, *131*, 1449–1466. [CrossRef]
5. Ekren, B.Y.; Sari, Z.; Lerher, T. Warehouse Design under Class-Based Storage Policy of Shuttle-Based Storage and Retrieval System. *IFAC-Pap.* **2015**, *48*, 1152–1154. [CrossRef]
6. Yang, P.; Yang, K.; Qi, M. Designing the optimal multi-deep AS/RS storage rack under full turnover-based storage policy based on non-approximate speed model of S/R machine. *Transp. Res. Part E Logist. Transp. Rev.* **2017**, *104*, 113–130. [CrossRef]
7. Ekren, B.Y.; Akpunar, A. An open queuing network-based tool for performance estimations in a shuttle-based storage and retrieval system. *Appl. Math. Model.* **2021**, *89*, 1678–1695. [CrossRef]
8. Lerher, T.; Ficko, M.; Palčič, I. Throughput performance analysis of Automated Vehicle Storage and Retrieval Systems with multiple-tier shuttle vehicles. *Appl. Math. Model.* **2021**, *91*, 1004–1022. [CrossRef]
9. Dong, W.; Jin, M. Travel time models for tier-to-tier SBS/RS with different storage assignment policies and shuttle dispatching rules. *Transp. Res. Part E Logist. Transp. Rev.* **2021**, *155*, 102485. [CrossRef]

10. Manzini, R.; Accorsi, R.; Baruffaldi, G.; Cennerazzo, T.; Gamberi, M. Travel time models for deep-lane unit-load autonomous vehicle storage and retrieval system (AVS/RS). *Int. J. Prod. Res.* **2016**, *54*, 4286–4304. [CrossRef]
11. D'Antonio, G.; Chiabert, P. Analytical models for cycle time and throughput evaluation of multi-shuttle deep-lane AVS/RS. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 1919–1936. [CrossRef]
12. Fragapane, G.; de Koster, R.; Sgarbossa, F.; Strandhagen, J.O. Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *Eur. J. Oper. Res.* **2021**, *294*, 405–426. [CrossRef]
13. Wan, J.; Tang, S.; Hua, Q.; Li, D.; Liu, C.; Lloret, J. Context-Aware Cloud Robotics for Material Handling in Cognitive Industrial Internet of Things. *IEEE Internet Things J.* **2017**, *5*, 2272–2281. [CrossRef]
14. Corréa, A.I.; Langevin, A.; Rousseau, L.M. Scheduling and routing of automated guided vehicles: A hybrid approach. *Comput. Oper. Res.* **2007**, *34*, 1688–1707. [CrossRef]
15. Ghasemzadeh, H.; Behrangi, E.; Azgomi, M.A. Conflict-free scheduling and routing of automated guided vehicles in mesh topologies. *Robot. Auton. Syst.* **2009**, *57*, 738–748. [CrossRef]
16. Fazlollahtabar, H.; Saidi-Mehrabad, M.; Balakrishnan, J. Mathematical optimization for earliness/tardiness minimization in a multiple automated guided vehicle manufacturing system via integrated heuristic algorithms. *Robot. Auton. Syst.* **2015**, *72*, 131–138. [CrossRef]
17. Bakshi, S.; Feng, T.; Yan, Z.; Chen, D. Fast Scheduling of Autonomous Mobile Robots under Task Space Constraints with Priorities. *J. Dyn. Syst. Meas. Control Trans. ASME* **2019**, *141*, 071009. [CrossRef]
18. De Ryck, M.; Versteyhe, M.; Debrouwere, F. Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *J. Manuf. Syst.* **2020**, *54*, 152–173. [CrossRef]
19. Liaqat, A.; Hutabarat, W.; Tiwari, D.; Tinkler, L.; Harra, D.; Morgan, B.; Taylor, A.; Lu, T.; Tiwari, A. Autonomous mobile robots in manufacturing: Highway Code development, simulation, and testing. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 4617–4628. [CrossRef]
20. Wu, N.; Zhou, M. Shortest routing of bidirectional automated guided vehicles avoiding deadlock and blocking. *Trans. Mechatron.* **2007**, *12*, 63–72. [CrossRef]
21. Saidi-Mehrabad, M.; Dehnavi-Arani, S.; Evazabadian, F.; Mahmoodian, V. An Ant Colony Algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs. *Comput. Ind. Eng.* **2015**, *86*, 2–13. [CrossRef]
22. Yang, Y.; Zhong, M.; Dessouky, Y.; Postolache, O. An integrated scheduling method for AGV routing in automated container terminals. *Comput. Ind. Eng.* **2018**, *126*, 482–493. [CrossRef]
23. Zhang, Z.; Guo, Q.; Chen, J.; Yuan, P. Collision-Free Route Planning for Multiple AGVs in an Automated Warehouse Based on Collision Classification. *IEEE Access* **2018**, *6*, 26022–26035. [CrossRef]
24. Digani, V.; Hsieh, M.A.; Sabattini, L.; Secchi, C. Coordination of multiple AGVs: A quadratic optimization method. *Auton. Robot.* **2019**, *43*, 539–555.
25. Li, K.; Liu, T.; Kumar, P.N.R.; Han, X. A reinforcement learning-based hyper-heuristic for AGV task assignment and route planning in parts-to-picker warehouses. *Transp. Res. Part E Logist. Transp. Rev.* **2024**, *185*, 103518.
26. Ho, G.T.S.; Tang, Y.M.; Leung, E.K.H.; Tong, P. Integrated reinforcement learning of automated guided vehicles dynamic path planning for smart logistics and operations. *Transp. Res. Part E Logist. Transp. Rev.* **2025**, *196*, 104008. [CrossRef]
27. Heik, D.; Bahrpeyma, F.; Reichelt, D. Study on the application of single-agent and multi-agent reinforcement learning to dynamic scheduling in manufacturing environments with growing complexity: Case study on the synthesis of an industrial IoT Test Bed. *J. Manuf. Syst.* **2024**, *77*, 525–557. [CrossRef]
28. Ha, T.; Chae, J. Free balancing for a shuttle-based storage and retrieval system. *Simul. Model. Pract. Theory* **2018**, *82*, 12–31. [CrossRef]
29. Song, J.; Yang, M.; Zhou, X. Scheduling optimization of Automated Storage and Retrieval System Based on Four-way Shuttles. In Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China, 13–16 October 2020; pp. 524–529.
30. Cheng, W.; Zhuan, W.; Yu, W. Techniques of Automatic Modeling for Four-Way-Shuttle Based Storage and Retrieval System Based on Flexsim. *MATEC Web Conf.* **2020**, *325*, 05003. [CrossRef]
31. Li, M.; Li, L.; Zhang, C.; Jiang, L.; Liu, H.; Lin, Z.; Wei, L. A Four-Way Shuttle Scheduling Method Based on Grey Wolf Algorithm. In Proceedings of the China Automation Congress (CAC), Xiamen, China, 25–27 November 2022; pp. 1778–1784.
32. Mao, J.; Cheng, J.; Li, X.; Zhao, H.; Lin, C. Modelling Analysis of a Four-Way Shuttle-Based Storage and Retrieval System on the Basis of Operation Strategy. *Appl. Sci.* **2023**, *13*, 3306. [CrossRef]
33. Mao, J.; Cheng, J.; Li, X.; Xu, S.; Wu, W. Research on scheduling optimization of four-way shuttle-based storage and retrieval systems. *Sci. Rep.* **2023**, *13*, 3999. [CrossRef] [PubMed]
34. Liu, Q.; Wang, C.; Li, X.; Gao, L. An improved genetic algorithm with modified critical path-based searching for integrated process planning and scheduling problem considering automated guided vehicle transportation task. *J. Manuf. Syst.* **2023**, *70*, 127–136. [CrossRef]

35. Wu, Z.; Zhang, X.; Li, L.; Zhang, Z.; Zhao, B.; Zhang, Y.; He, X. Research on Inbound Jobs' Scheduling in Four-Way-Shuttle-Based Storage System. *Processes* **2024**, *12*, 223. [CrossRef]
36. Xu, L.; Liu, C.; Zhan, X.; Ling, X. Task Scheduling for Tier-to-tier Four-way Shuttle Warehousing System. *Procedia CIRP* **2021**, *104*, 1083–1088. [CrossRef]
37. Ren, C.; Lu, J.; Shao, Y.; Zhan, Y.; Jiang, W.; Zhao, W. Scheduling optimization of tier-captive four-way shuttle-based storage and retrieval systems with multiple lifts. *Eng. Optim.* **2024**, *57*, 459–477. [CrossRef]
38. Ren, C.; Yan, Q.; Liu, Z. Scheduling optimization in a multi-deep tier-to-tier four-way shuttle storage and retrieval system. *Comput. Ind. Eng.* **2025**, *204*, 111095. [CrossRef]
39. Han, X.; Tan, Y.; Chen, S.; Liu, Z.; Wang, H. Cooperative Path Planning for Four-Way Shuttle Vehicles in Storage and Retrieval Systems: A Hierarchically Dynamic Graph-Based Approach. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024; pp. 10275–10280.
40. Ding, S.; Liu, H.; Xu, S.; Wu, W. Research on Scheduling Optimization of Multi-layer Four-way Shuttle System Based on Composite Tasks. In Proceedings of the 36th Chinese Control and Decision Conference (CCDC), Xi'an, China, 25–27 May 2024; pp. 3765–3770.
41. Zhang, Z.; Wang, J.; Jin, J.; Wu, Z.; Wu, L.; Peng, T.; Li, P. Research on Task Allocation in Four-Way Shuttle Storage and Retrieval Systems Based on Deep Reinforcement Learning. *Sustainability* **2025**, *17*, 6772. [CrossRef]